

Aspects of Correlation Matrices

Jan de Leeuw

May 25, 2015

Suppose we have a dataframe with n observations on m variables, i.e. m vectors in $\mathbb{R}^n$. We want to transform the m variables in such a way that a particular *aspect* of the correlation matrix of the variables is maximized. An *aspect* is a real valued function f defined on the space of correlation matrices.

We formalize this using the framework discussed in De Leeuw (1988a; 1988b). Variable j defines a convex cone $\mathcal{K}_j$ in $\mathbb{R}^n$. We suppose all vectors in the cone $\mathcal{K}_j$ are centered. Also define the sphere S of all vectors in $\mathbb{R}^n$ with sum of squares equal to one. For any choice of the $x_j \in \mathcal{K}_j \cap S$ we can compute the correlation matrix $R(X) = \{r_{j\ell}(X)\} = x_j' x_\ell$. Our optimization problem is to maximize $f(R(X))$ over all $x_j \in \mathcal{K}_j \cap S$.

In our software we define the cones $\mathcal{K}_j$ as the sets of all vectors x such that $x = Zb$ and (optionally) $Ax \geq 0$, where Z and A are given matrices. In Mair and De Leeuw (2010) the `R` package `aspect` is presented. It deals with the special case where the variables are categorical and the Z are *indicator matrices* (a.k.a. *dummies*). On page 20 in Mair and De Leeuw (2010) it says:

Considering each variable as categorical is not very efficient when having many categories, as typically in the numerical case. Therefore, in a future update we will use splines to make it more efficient.

Well, this is that future update. In this note we develop `R` software that handles the much more general case where the Z are B-spline bases. The inequality constraints can be used to require that the spline transformations in Zb are monotone with the data. This covers the categorical case, because spline bases of degree zero are indicator matrices. It also covers polynomial transformations by using splines without interior knots. And finally our software also allows for *nominal* variables, by not imposing monotonicity restrictions.

We now use a combination of tangential majorization and block relaxation (De Leeuw 2015a, De Leeuw (2015b)) to construct an algorithm maximizing f . Consider the case in which f is convex on the convex set of correlation matrices. Then

$$f(R(X)) \geq f(R(Y)) + \text{tr } G(Y)(R(X) - R(Y)),$$

where $G(Y) := Df(R(Y))$. In a majorization iteration we optimize $\text{tr } G(Y)R(X)$. For this we use cyclic block relaxation, scaling one variable at a time while keeping the others fixed at their current values. Because $\text{diag}(R(X)) = \text{diag}(R(Y))$ we solve subproblem j in the cycle by optimizing $x_j' t_j$ over $x_j \in \mathcal{K}_j \cap S$, where t_j is the *target*

$$t_j := \sum_{\ell \neq j}^m g_{j\ell}(Y) x_\ell.$$

Of course t_j depends on Y and the x_ℓ with $\ell \neq j$. We find the new x_j by minimizing $(t_j - Z_j b_j)'(t_j - Z_j b_j)$ over $A_j Z_j b_j \geq 0$ and by projecting the optimal solution $\hat{x}_j = Z_j \hat{b}_j$ on the sphere $\mathcal{S}$. We assume throughout that $\hat{x}_j$ is non-zero, i.e. the target is not in the polar cone.

In our R implementation we have incorporated some of the more obvious aspect choices that are convex functions of the correlation matrix: the sum of the dominant p eigenvalues (principal component analysis), the squared multiple correlation of one variable with the others (multiple regression), the sum of all squared multiple correlations of one variable with the others (image analysis), the negative logarithm of the determinant, the sum of p th powers of the correlations (with suitable p), the sum of p th powers of the absolute values of the correlations. Thus our implementation covers both non-linear principal component analysis (Young, Takane, and Leeuw 1978, Winsberg and Ramsay (1983), Koyak (1987), Linting et al. (2007), De Leeuw (2014)) and multiple regression with optimal scaling (Kruskal 1965, Young, Leeuw, and Takane (1976), Winsberg and Ramsay (1980), Van Der Kooij, Meulman, and Heiser (2007)). And much more, obviously.

Code

```
aspect <-
  function (data, knots, degrees, ordinal, afunc, eps = 1e-6, itmax = 100, verbose =
1, ...) {
  m <- ncol (data)
  n <- nrow (data)
  itel <- 1
  bd <- basesData (data, knots, degrees, ordinal)
  tdata <- matrix (0, n, m)
  for (j in 1:m) {
    tdata[,j] <- bd$x[[j]]
  }
  tdata <- apply (tdata, 2, function (z)
    z - mean (z))
  tdata <- apply (tdata, 2, function (z)
    z / sqrt (sum (z ^ 2)))
  corr <- crossprod (tdata)
  af <- afunc(corr, ...)
  fold <- af$f
  g <- af$g
  repeat {
    for (j in 1:m) {
      target <- drop (tdata[,-j] %*% g[-j, j])
      k <- bd$b[[j]]
      v <- bd$v[[j]]
      u <- drop (crossprod(k, target))
      s0 <- sum(target * tdata[,j])
      if (ordinal[j]) {
        ns <- nnls(v,u)
        rr <- residuals(ns)
        tt <- drop(k %*% rr)
      } else
        tt <- drop (k %*% u)
      tt <- tt - mean (tt)
```

```

sq <- sum(tt ^ 2)
if (sq > 1e-15) {
  tt <- tt / sqrt (sum (tt ^ 2))
  tdata[,j] <- tt
}
s1 <- sum(target * tdata[,j])
if (verbose > 1)
  cat (
    "**** Variable", formatC(j, width = 3, format = "d"),
    "Before", formatC(
      s0, digits = 8, width = 12, format = "f"
    ),
    "After", formatC(
      s1, digits = 8, width = 12, format = "f"
    ),
    "\n"
  )
corr <- crossprod (tdata)
af <- afunc (corr, ...)
fnew <- af$f
g <- af$g
}
if (verbose > 0)
  cat(
    "Iteration: ", formatC (itel, width = 3, format = "d"),
    "fold: ", formatC (
      fold, digits = 8, width = 12, format = "f"
    ),
    "fnew: ", formatC (
      fnew, digits = 8, width = 12, format = "f"
    ),
    "\n"
  )
if ((itel == itemax) || ((fnew - fold) < eps))
  break
itel <- itel + 1
fold <- fnew
}
return (list (
  tdata = tdata, f = fnew, r = corr, g = g
))
}

```

The program uses the `nnls` package (Mullen and Stokkum 2012) for the inequality restricted regressions, with the dual algorithm described by De Leeuw (2015c). The inequality constraints always require monotonicity of the transformed values Zb with the original data.

The function `aspect` starts out by calling the subroutine `basesData`, which constructs and stores orthonormalized B-spline bases, initial transformation estimates, and other auxiliary quantities in lists. The subroutine uses an ad-hoc B-spline implementation written in `c` (De Leeuw 2015d), and a similarly ad-hoc

Gram-Schmidt in `c` (De Leeuw 2015e). For reproducibility purposes the `c` code is incorporated in this document. Note that we do not use monotone splines, such as the integrated B-splines of Winsberg and Ramsay (1980, @winsberg_ramsay_83), we merely require that our splines are monotone at the data points.

```
basesData <- function (data, knots, degrees, ordinal) {
  m <- ncol (data)
  n <- nrow (data)
  b <- as.list (rep (0, m))
  v <- as.list (rep (0, m))
  x <- as.list (rep (0, m))
  for (j in 1:m) {
    b[[j]] <- bsplineBasis(data[[j]], degrees[j], knots[[j]])
    nb <- ncol(b[[j]])
    x[[j]] <- drop (b[[j]] %*% (1 : nb))
    b[[j]] <- apply (b[[j]], 2, function (z)
      z - mean (z))
    b[[j]] <- (gsrsrc (b[[j]])$q)[,1 : (nb - 1), drop=FALSE]
    if (ordinal[j]) {
      v[[j]] <- matrix (0, nb - 1, n - 1)
      r <- rank (data[[j]], ties.method = "first")
      for (i in 1 : (n - 1)) {
        v[[j]][, i] <- b[[j]][which(r == i),] - b[[j]][which(r == (i + 1)),]
      }
    }
  }
  return (list (b = b, v = v, x = x))
}
```

Here are the implementations of various aspects. Of course the user can add her own aspect by writing a similar function.

```
maxeig <- function (r, p) {
  e <- eigen (r)
  f <- sum (e$values[1:p])
  g <- tcrossprod(e$vectors[,1:p])
  return (list (f = f, g = g))
}

maxcor <- function (r, p) {
  f <- sum (r ^ p)
  g <- p * (r ^ (p - 1))
  return (list (f = f, g = g))
}

maxabs <- function (r, p) {
  f <- sum (abs(r)^ p)
  g <- p * (abs(r) ^ (p - 1))*sign(r)
  return (list (f = f, g = g))
}
```

```

}

maxdet <- function (r) {
  f <- -log(det (r))
  g <- -solve(r)
  return (list (f = f, g = g))
}

maxsmc <- function (r, p) {
  beta <- solve (r[-p, -p], r[p, -p])
  f <- sum (beta * r[p, -p])
  h <- rep (1, nrow (r))
  h[-p] <- -beta
  g <- -outer (h, h)
  return (list (f = f, g = g))
}

maxsum <- function (r, p) {
  f <- sum (sqrt (r ^ 2 + p))
  g <- r / sqrt (r ^ 2 + p)
  return (list (f = f, g = g))
}

maximage <- function (r) {
  n <- nrow(r)
  f <- 0
  g <- matrix (0, n, n)
  for (p in 1:n) {
    beta <- solve (r[-p,-p], r[p,-p])
    f <- f + sum (beta * r[p,-p])
    h <- rep (1, nrow (r))
    h[-p] <- -beta
    g <- g - outer (h, h)
  }
  return (list (f = f, g = g))
}

```

Example 1: Gibbs-Neumann Data

Our first example are the Neumann data used by Willard Gibbs in his study of mixtures of gases with convertible components. These data have also been analyzed in Gifi (1990, 370–75), where there is more information about them. We choose the three knots at the quartiles.

```

data(neumann, package = "homals")
neumann_knots <- lapply (neumann, function (x) fivenum (x)[2:4])
neumann_degrees <- c(0,2,2)
neumann_ordinal <-c(FALSE, TRUE, TRUE)

```

First we maximize the multiple correlation of the last variable (density) with the first two (temperature and pressure). Temperature is scaled as nominal and categorical, pressure and density are ordinal with spline degree 2.

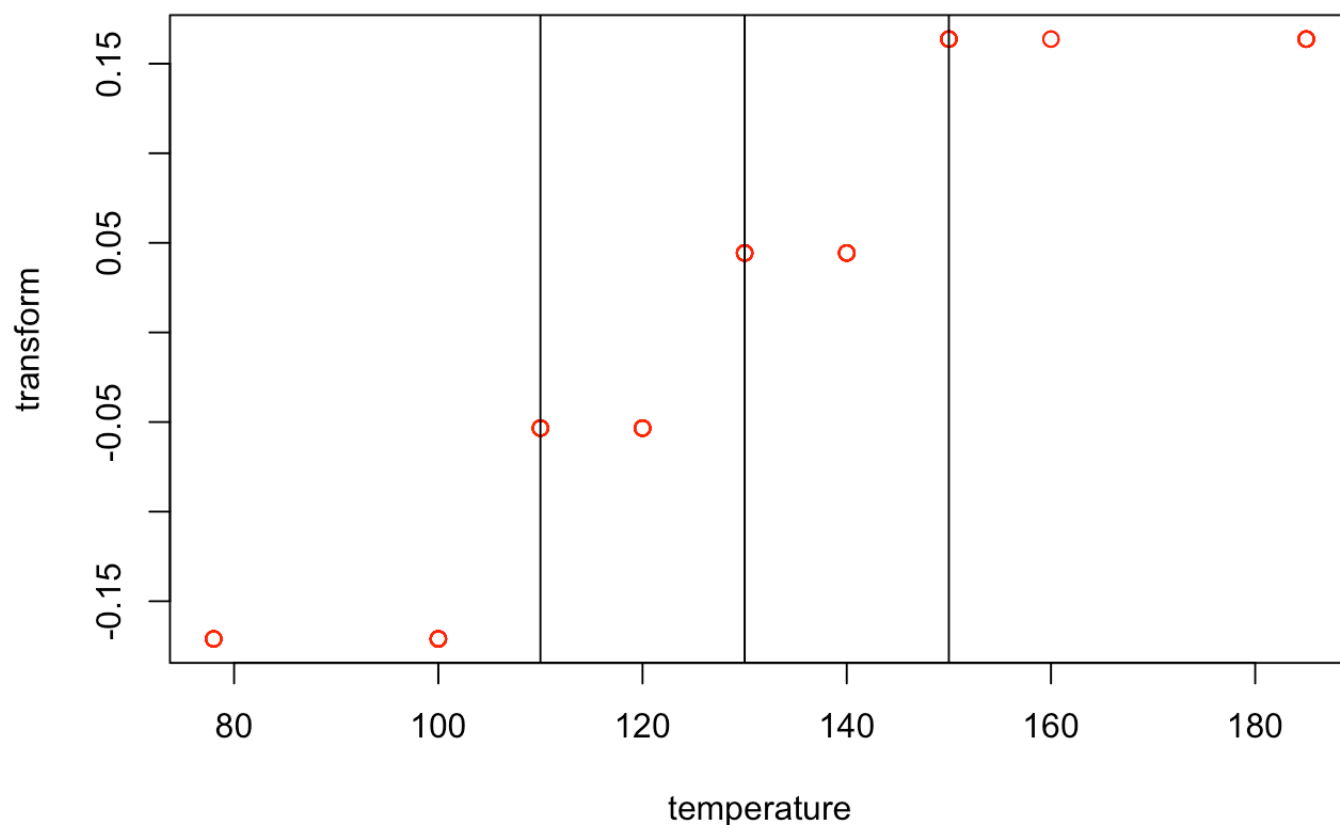
```
hreg<-aspect(neumann,neumann_knots,neumann_degrees,neumann_ordinal,maxsmc,p=3)
```

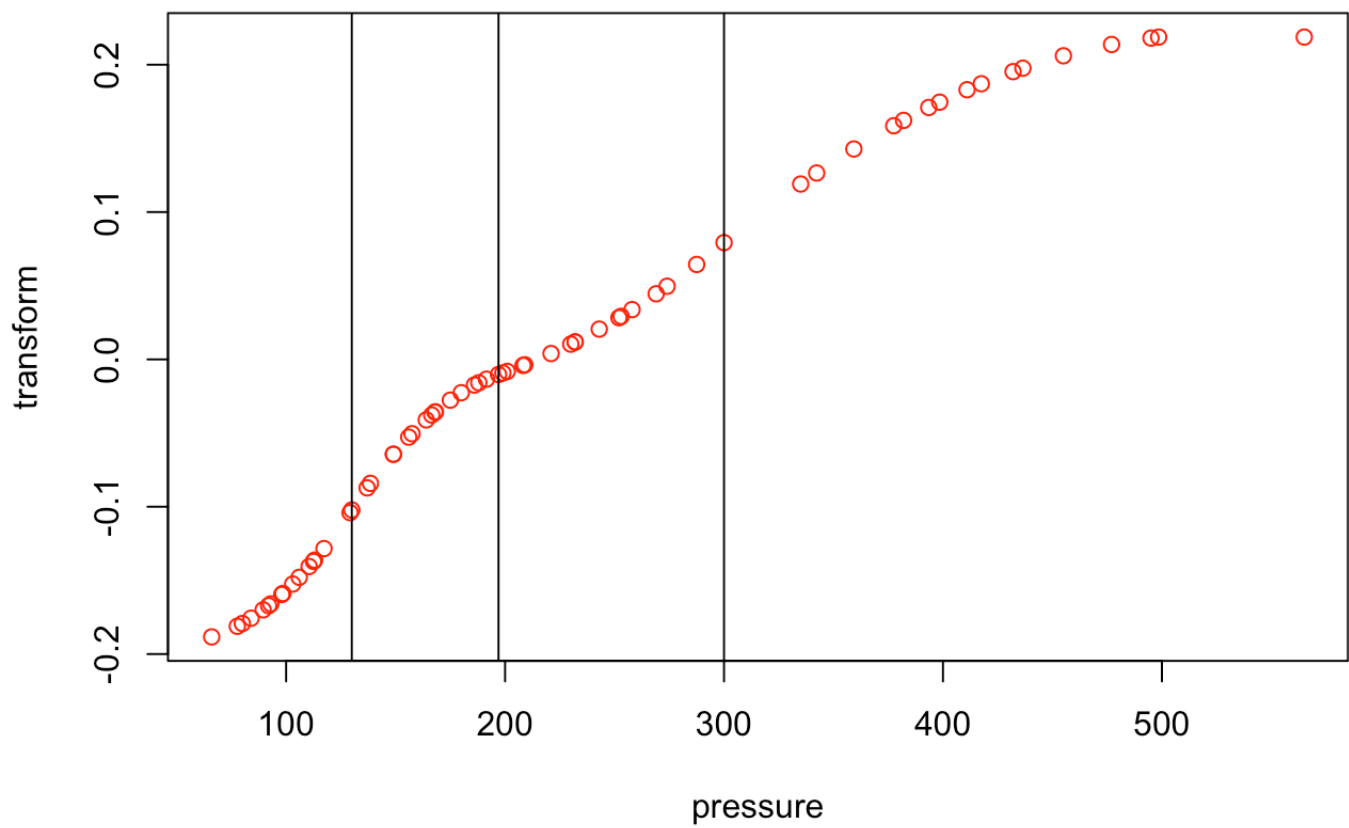
```
## Iteration:    1 fold:    0.88154151 fnew:    0.89539023
## Iteration:    2 fold:    0.89539023 fnew:    0.89566693
## Iteration:    3 fold:    0.89566693 fnew:    0.89566976
## Iteration:    4 fold:    0.89566976 fnew:    0.89567005
```

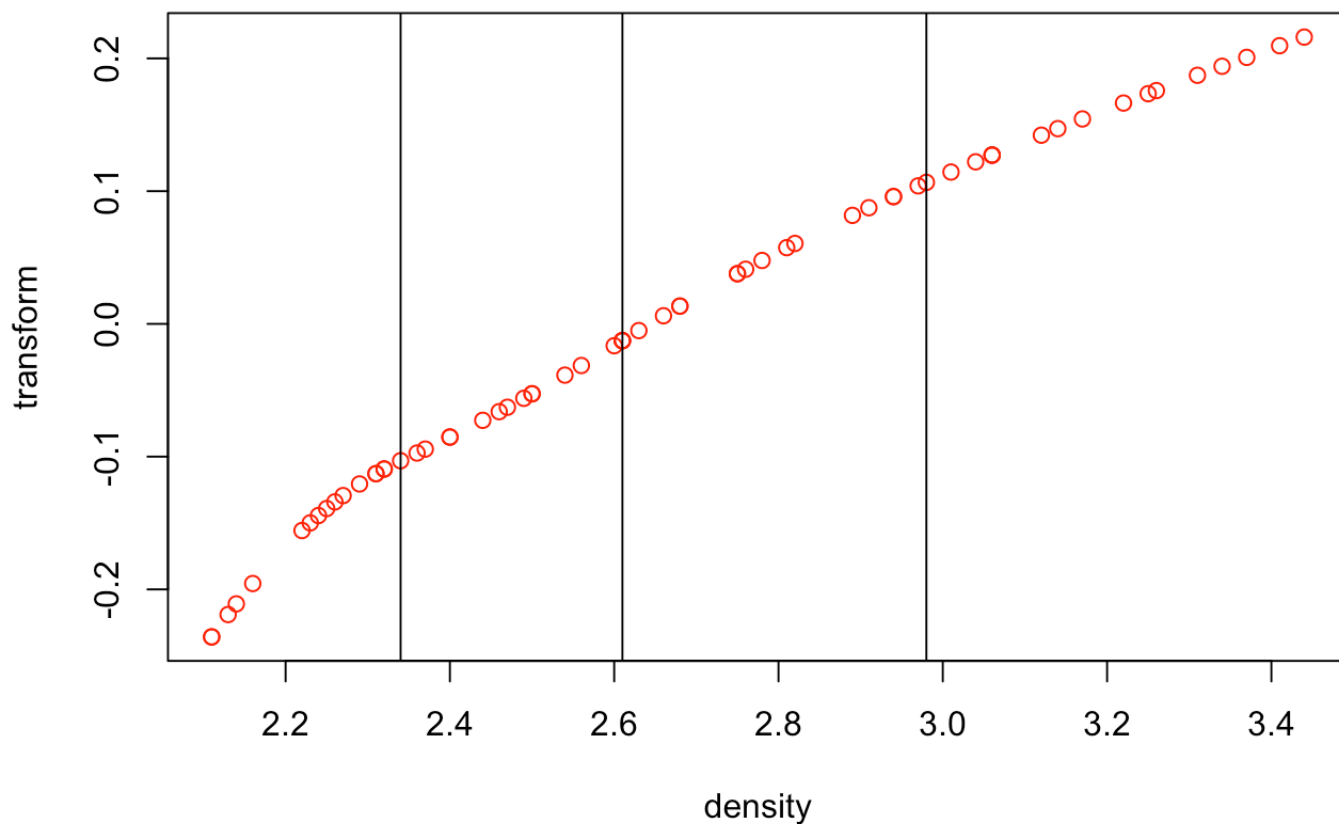
This gives the correlation matrix

```
##           [,1]      [,2]      [,3]
## [1,]  1.0000000  0.3169096 -0.8141950
## [2,]  0.3169096  1.0000000  0.1995548
## [3,] -0.8141950  0.1995548  1.0000000
```

and the optimal transformations (with vertical lines at the knots)







Next we compute a one-dimensional PCA by maximizing the largest eigenvalue (using the same knots and degrees).

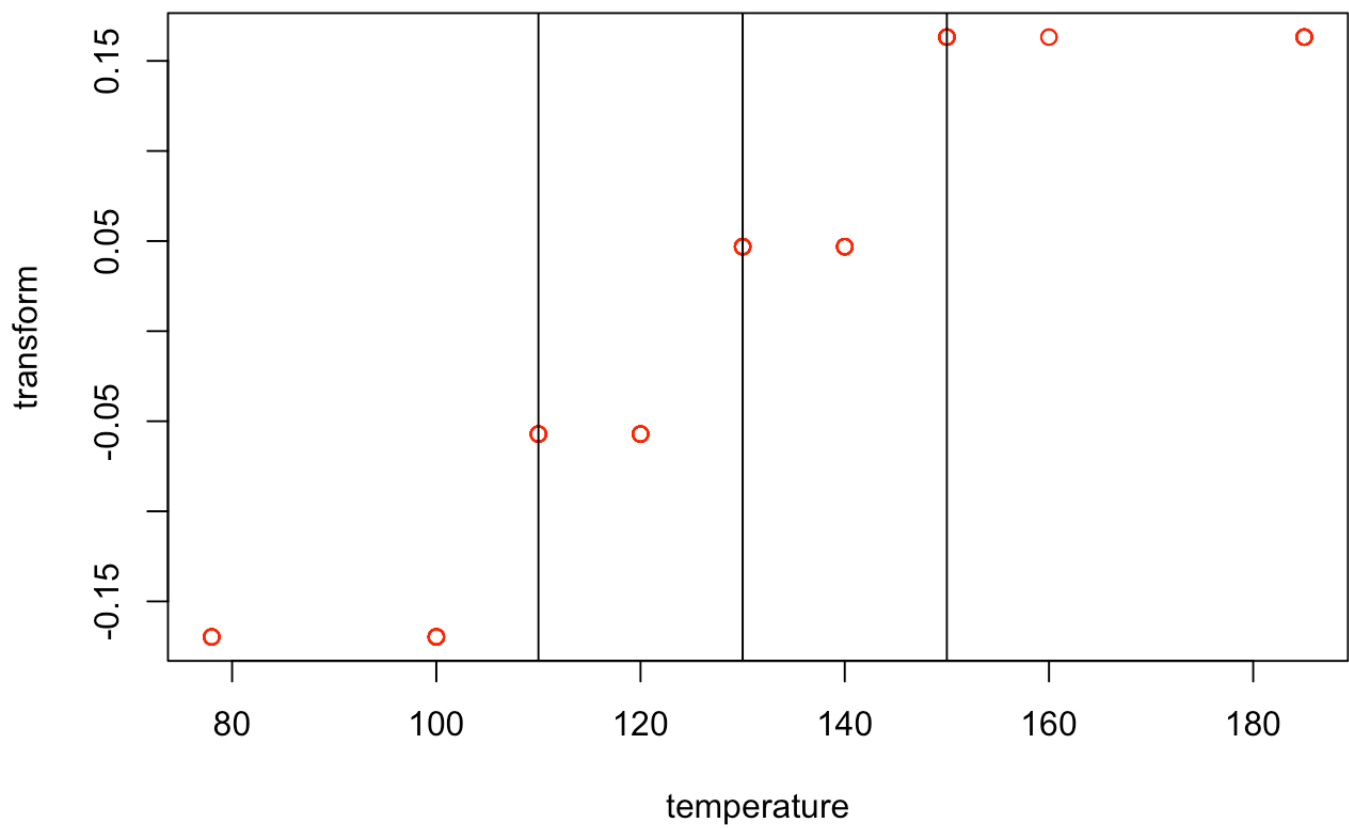
```
hpca<-aspect(neumann,neumann_knots,neumann_degrees,neumann_ordinal,maxeig,p=1)
```

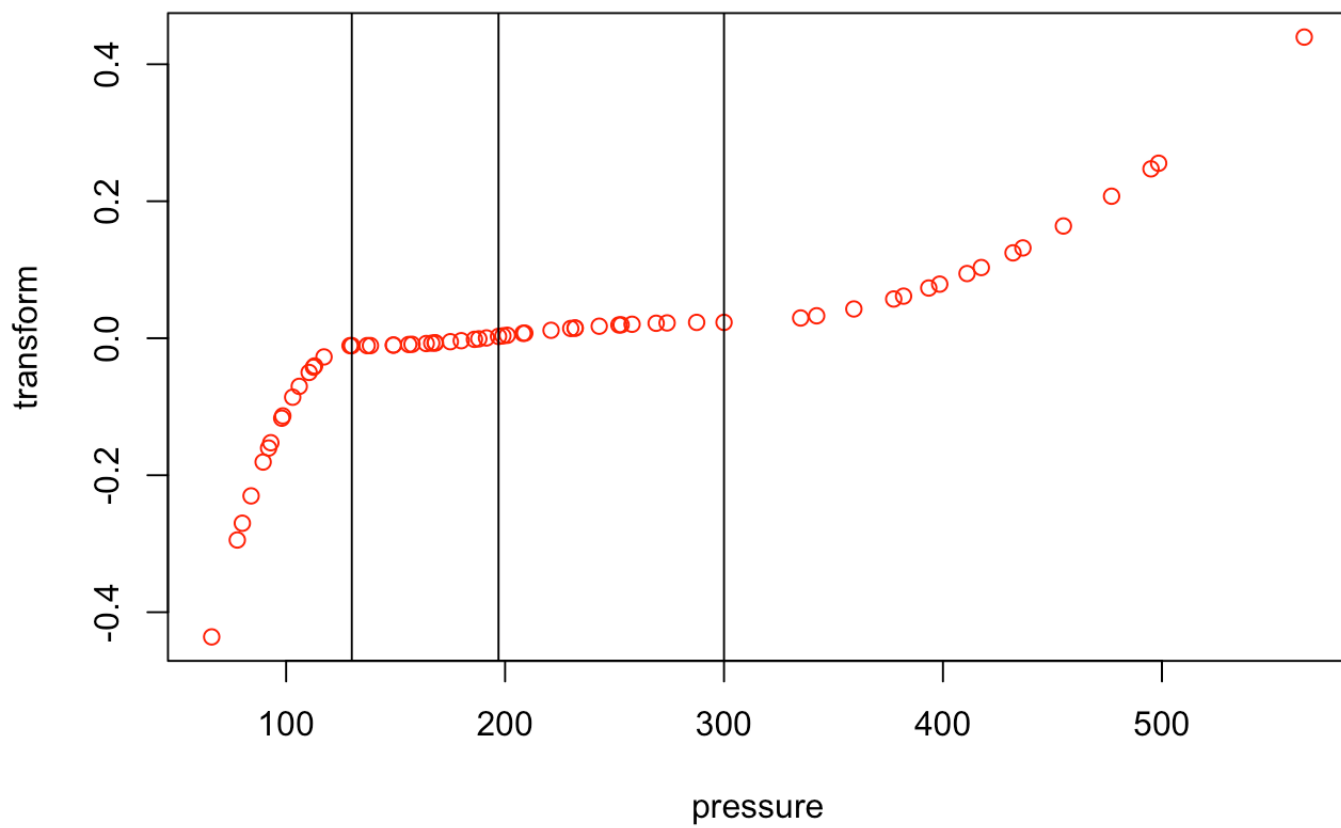
```
## Iteration:    1 fold:    1.83957300 fnew:    1.91034190
## Iteration:    2 fold:    1.91034190 fnew:    1.91058869
## Iteration:    3 fold:    1.91058869 fnew:    1.91059236
## Iteration:    4 fold:    1.91059236 fnew:    1.91059268
```

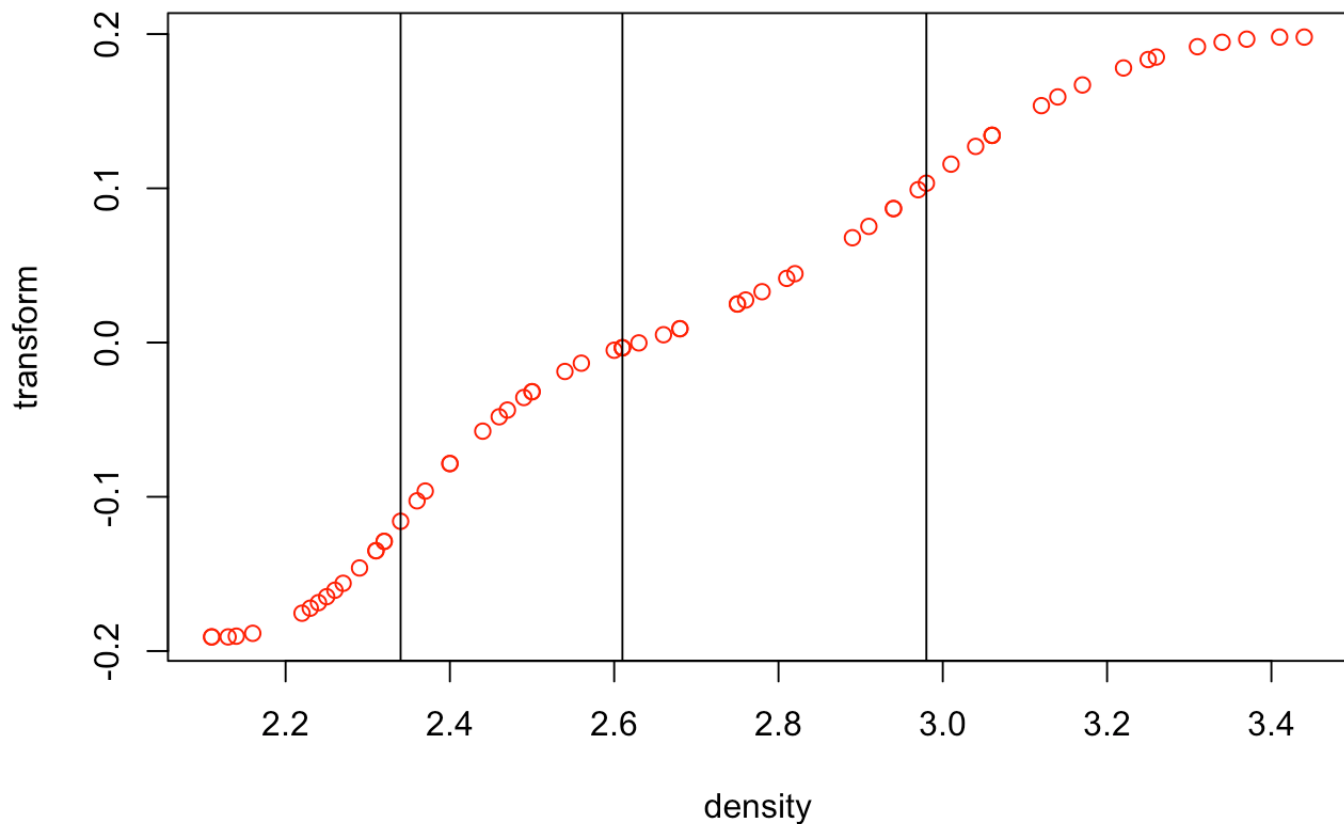
The correlation matrix now is

```
##           [,1]      [,2]      [,3]
## [1,]  1.0000000  0.400810296 -0.819241802
## [2,]  0.4008103  1.000000000  0.003661109
## [3,] -0.8192418  0.003661109  1.000000000
```

and the optimal transformations are







Next, we maximize what is perhaps the simplest aspect: the sum of the correlations.

```
hsum<-aspect(neumann,neumann_knots,neumann_degrees,neumann_ordinal,maxcor,p=1)
```

```
## Iteration:    1 fold:    2.40147297 fnew:    4.20649363
## Iteration:    2 fold:    4.20649363 fnew:    4.20663743
## Iteration:    3 fold:    4.20663743 fnew:    4.20669120
## Iteration:    4 fold:    4.20669120 fnew:    4.20671222
## Iteration:    5 fold:    4.20671222 fnew:    4.20672045
## Iteration:    6 fold:    4.20672045 fnew:    4.20672367
## Iteration:    7 fold:    4.20672367 fnew:    4.20672493
## Iteration:    8 fold:    4.20672493 fnew:    4.20672543
```

This gives the correlation matrix

```
hsum$r
```

```
##           [,1]      [,2]      [,3]
## [1,]  1.0000000 -0.3596036  0.8030536
## [2,] -0.3596036  1.0000000  0.1599128
## [3,]  0.8030536  0.1599128  1.0000000
```

The sum of the absolute values of the correlations.

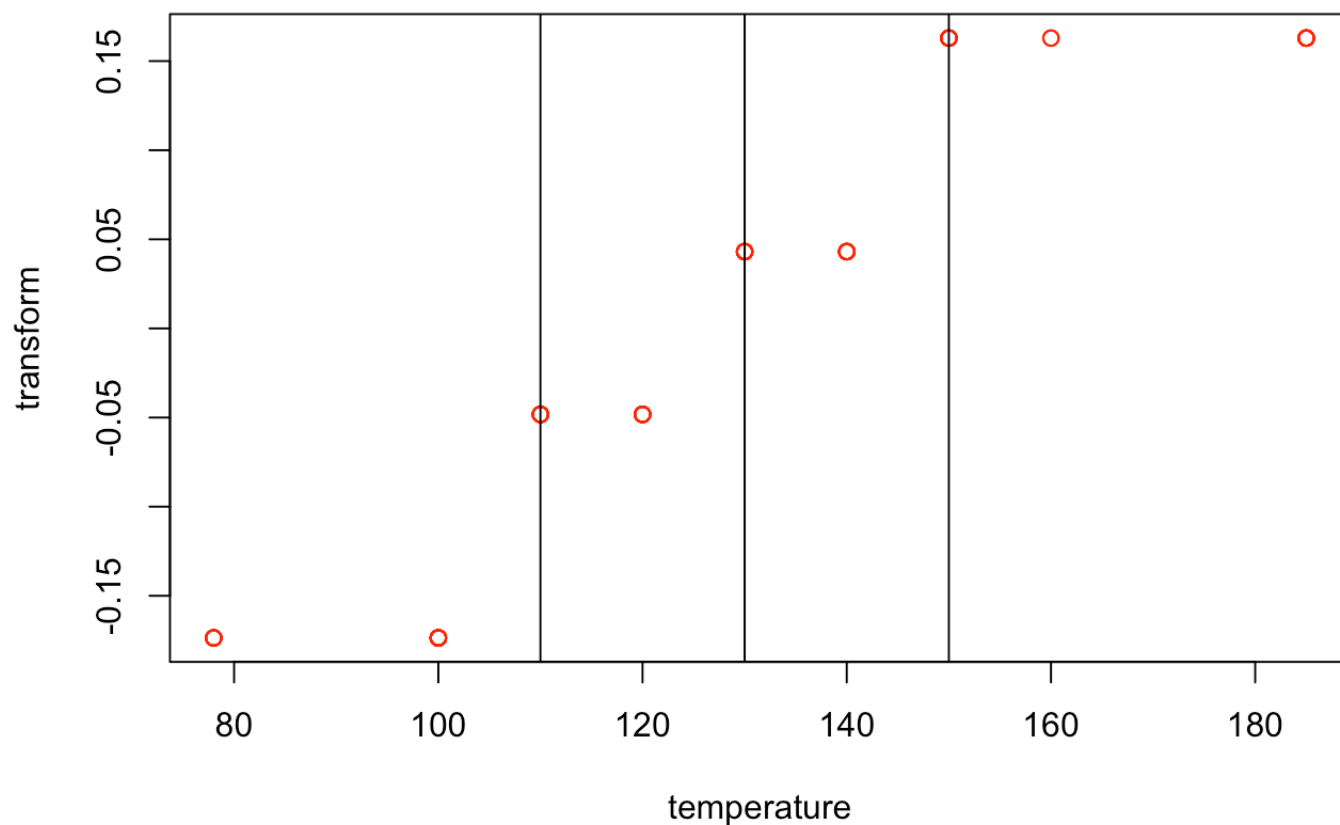
```
habs<-aspect(neumann,neumann_knots,neumann_degrees,neumann_ordinal,maxabs,p=1)
```

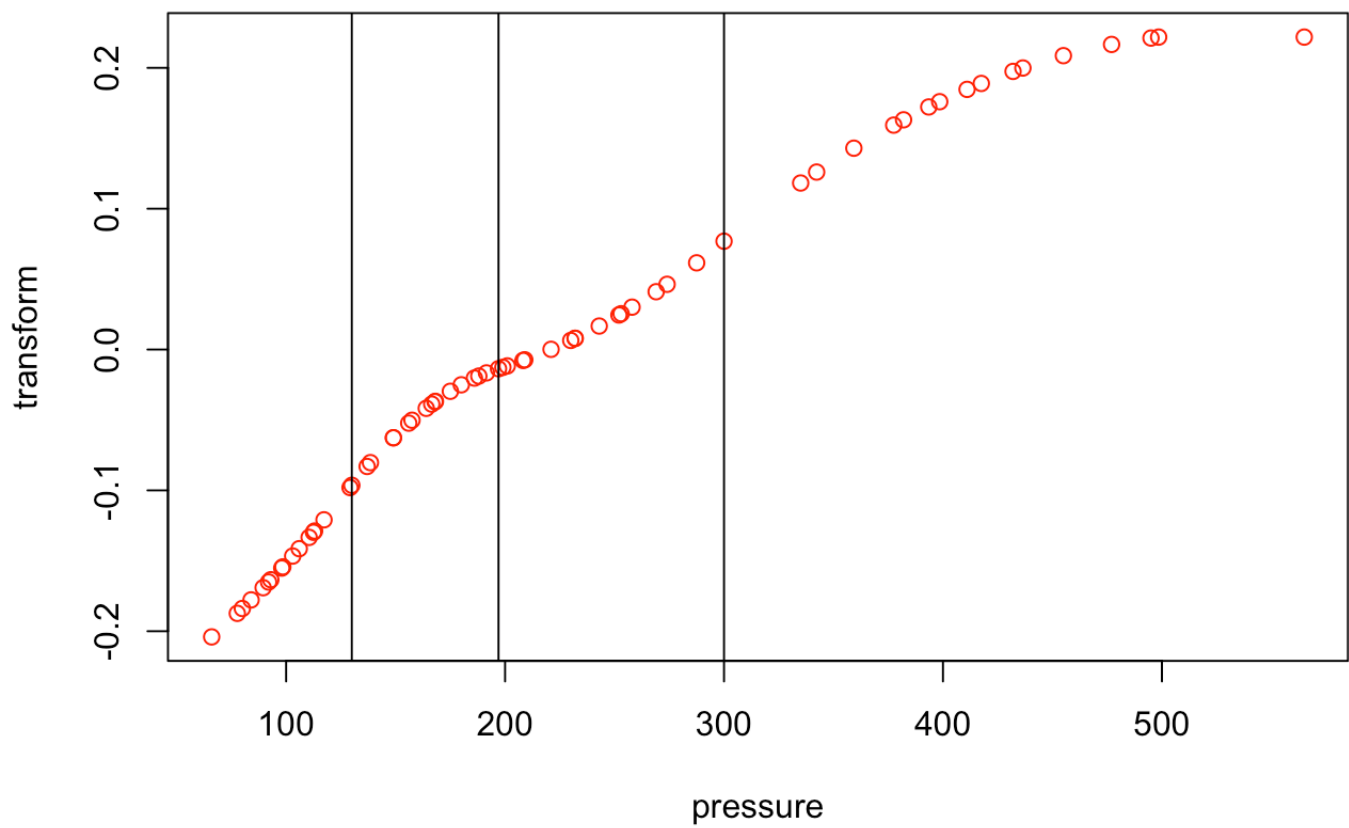
```
## Iteration:    1 fold:    5.64234387 fnew:    5.66909477
## Iteration:    2 fold:    5.66909477 fnew:    5.66992570
## Iteration:    3 fold:    5.66992570 fnew:    5.66996738
## Iteration:    4 fold:    5.66996738 fnew:    5.66997382
## Iteration:    5 fold:    5.66997382 fnew:    5.66997485
## Iteration:    6 fold:    5.66997485 fnew:    5.66997501
```

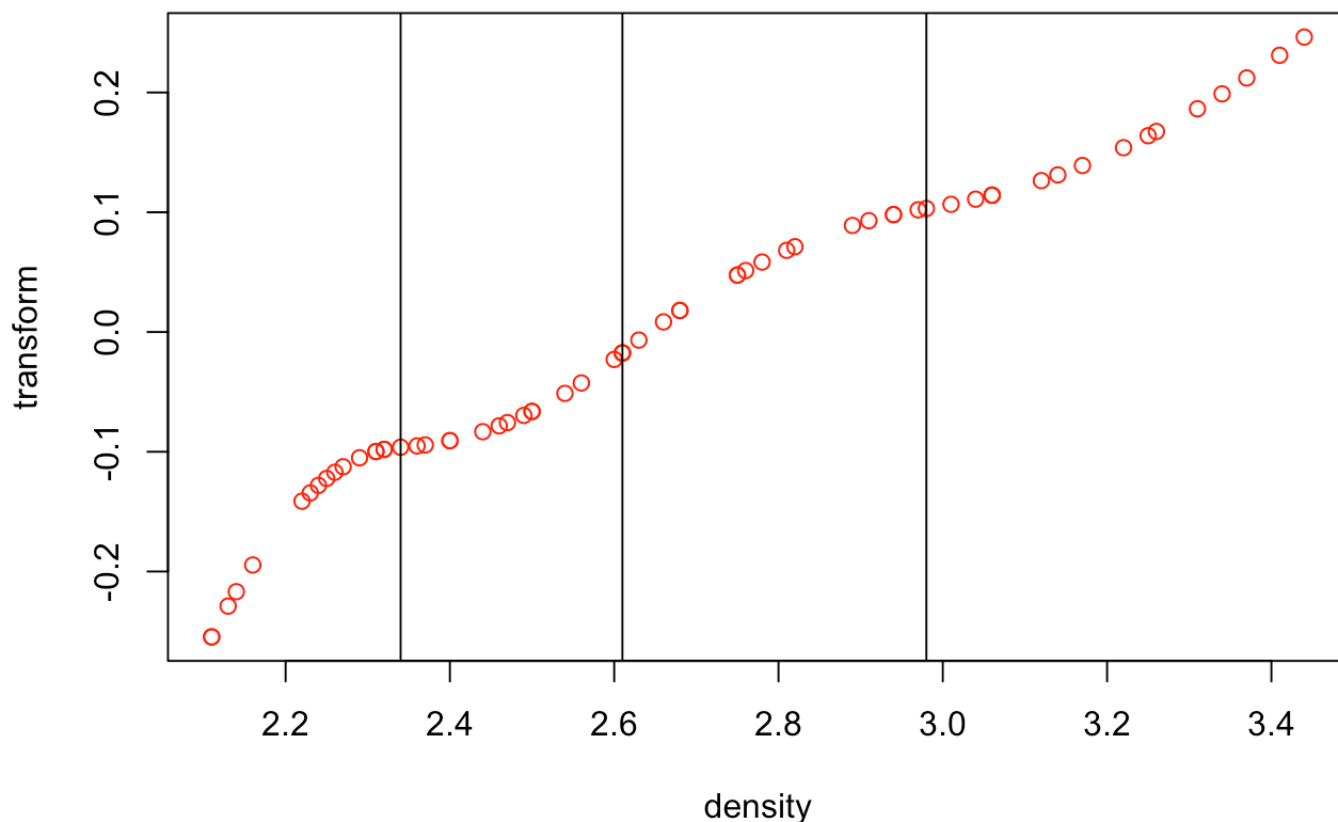
In this case the correlation matrix after the last iteration is

```
##           [,1]      [,2]      [,3]
## [1,]  1.0000000  0.3209786 -0.8013430
## [2,]  0.3209786  1.0000000  0.2126659
## [3,] -0.8013430  0.2126659  1.0000000
```

and the optimal transformations are







Example 2: Bodyfat Data

Age, weight, height, and 10 body circumference measurements are recorded for 252 men. Each man's percentage of body fat was accurately estimated by an underwater weighing technique.

```
data(fat, package = "faraway")
bodyfat <- fat
bodyfat_knots <- lapply (bodyfat, function (x) fivenum (x)[2:4])
bodyfat_degrees <- rep(2, 18)
bodyfat_ordinal <- rep (TRUE, 18)
bodyfat_ordinal[4]<-FALSE
names(bodyfat)
```

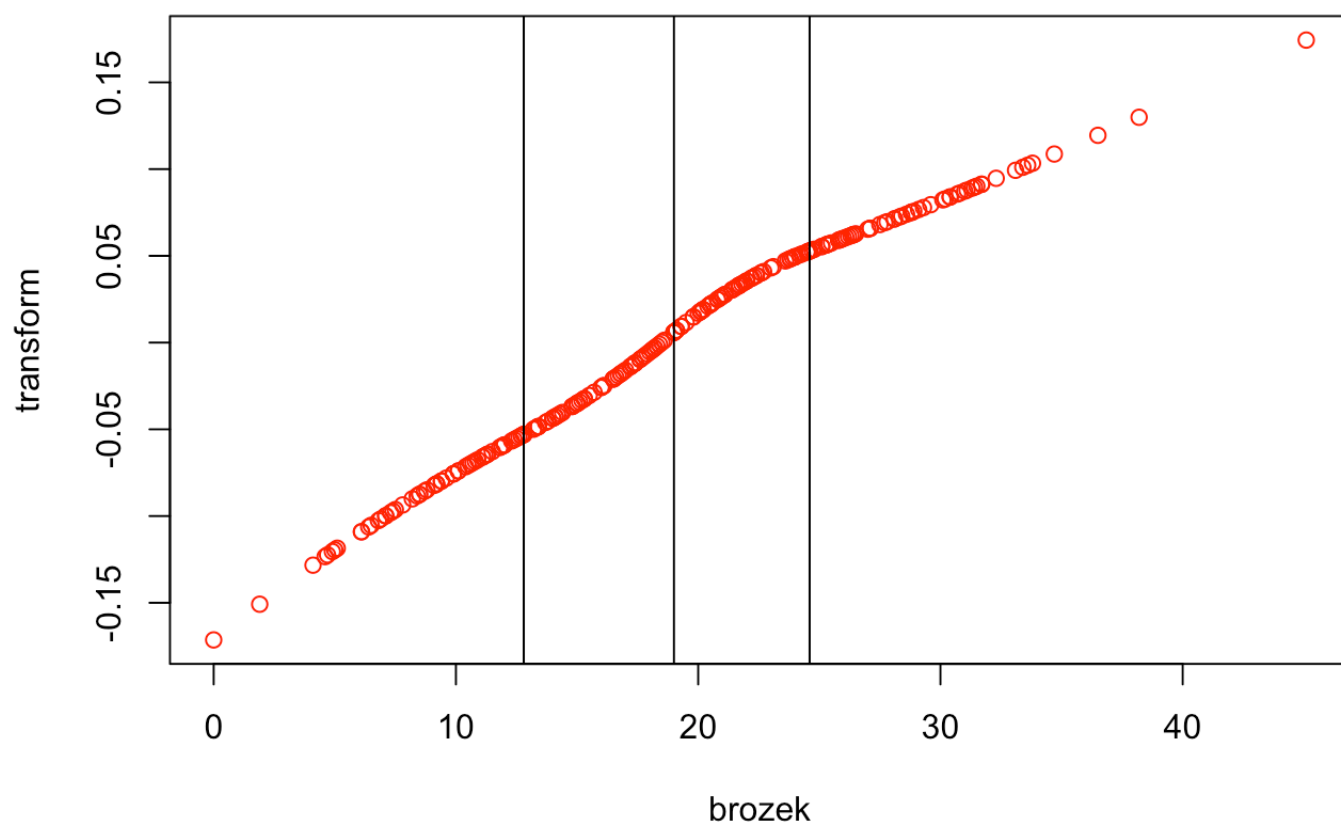
```
## [1] "brozek" "siri" "density" "age" "weight" "height" "adipos"
## [8] "free" "neck" "chest" "abdom" "hip" "thigh" "knee"
## [15] "ankle" "biceps" "forearm" "wrist"
```

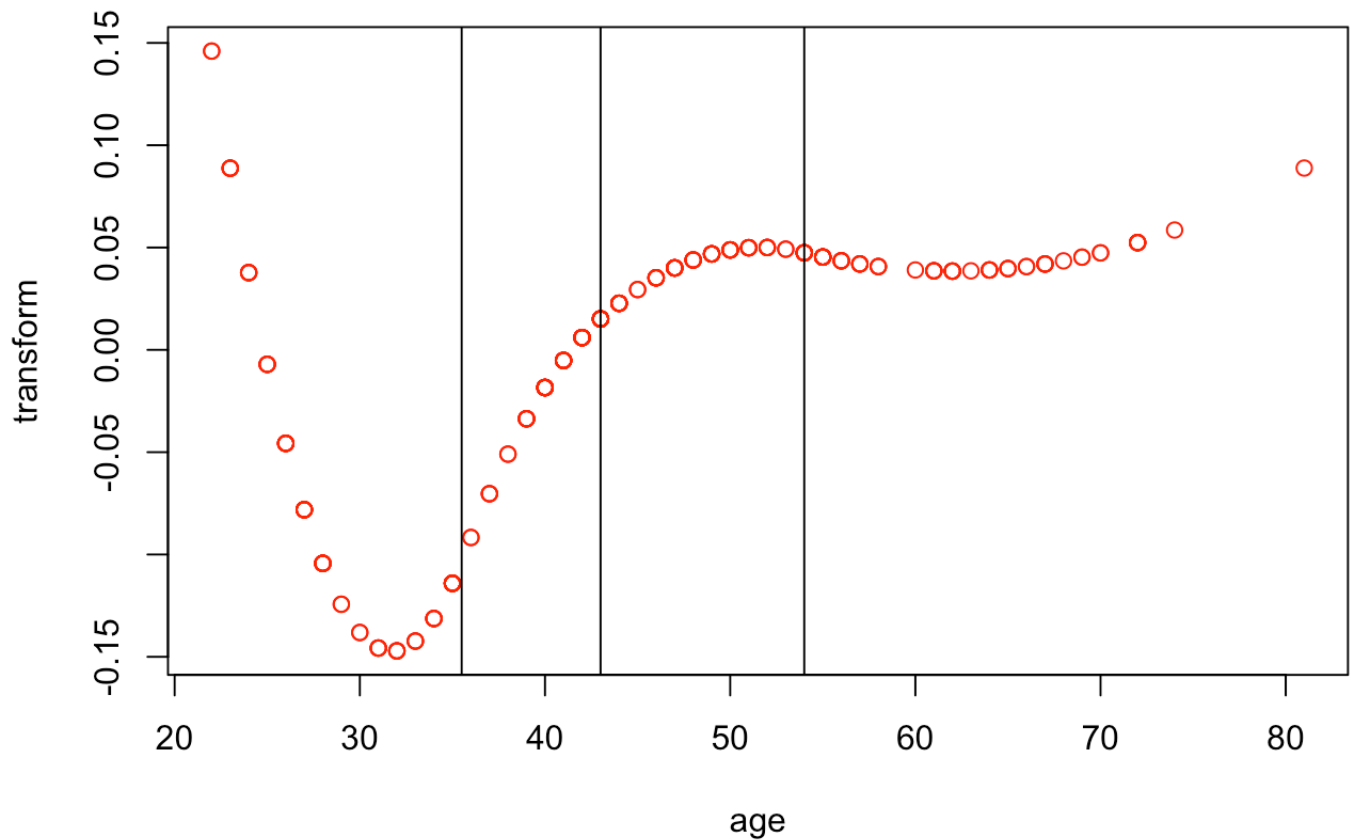
The first two variables are bodyfat indices, computed with different formulas. We see how well we can predict the first one using the multiple correlation aspect. All variables are transformed monotonically, except age.

```
bsmc<-aspect(bodyfat,bodyfat_knots,bodyfat_degrees,bodyfat_ordinal,maxsmc,p=1)
```

```
## Iteration: 1 fold: 0.99937103 fnew: 0.99971962
## Iteration: 2 fold: 0.99971962 fnew: 0.99974700
## Iteration: 3 fold: 0.99974700 fnew: 0.99975850
## Iteration: 4 fold: 0.99975850 fnew: 0.99976400
## Iteration: 5 fold: 0.99976400 fnew: 0.99976708
## Iteration: 6 fold: 0.99976708 fnew: 0.99976926
## Iteration: 7 fold: 0.99976926 fnew: 0.99977107
## Iteration: 8 fold: 0.99977107 fnew: 0.99977264
## Iteration: 9 fold: 0.99977264 fnew: 0.99977406
## Iteration: 10 fold: 0.99977406 fnew: 0.99977537
## Iteration: 11 fold: 0.99977537 fnew: 0.99977659
## Iteration: 12 fold: 0.99977659 fnew: 0.99977775
## Iteration: 13 fold: 0.99977775 fnew: 0.99977885
## Iteration: 14 fold: 0.99977885 fnew: 0.99977990
## Iteration: 15 fold: 0.99977990 fnew: 0.99978090
## Iteration: 16 fold: 0.99978090 fnew: 0.99978184
```

We plot the optimal transformation for the brozek index and for age.





We can use nonlinear principal component analysis to look more closely at the structure of the variables defining the indices. Optimize the sum of the two largest eigenvalues of the correlation matix of the last 16 variables.

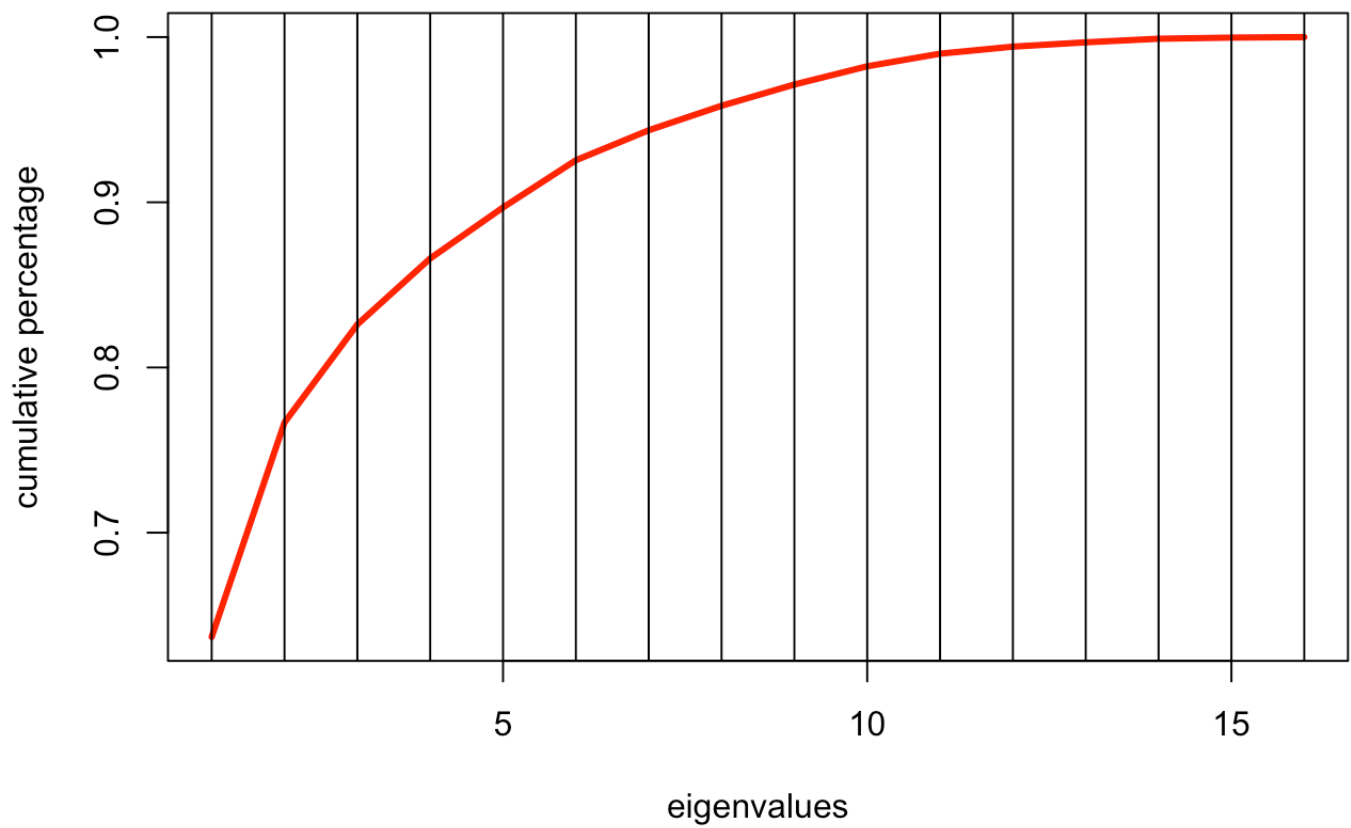
```
beig<-aspect(bodyfat[,3:18], bodyfat_knots[3:18], bodyfat_degrees[3:18], bodyfat_ordinal[3:18], max eig, p = 2)
```



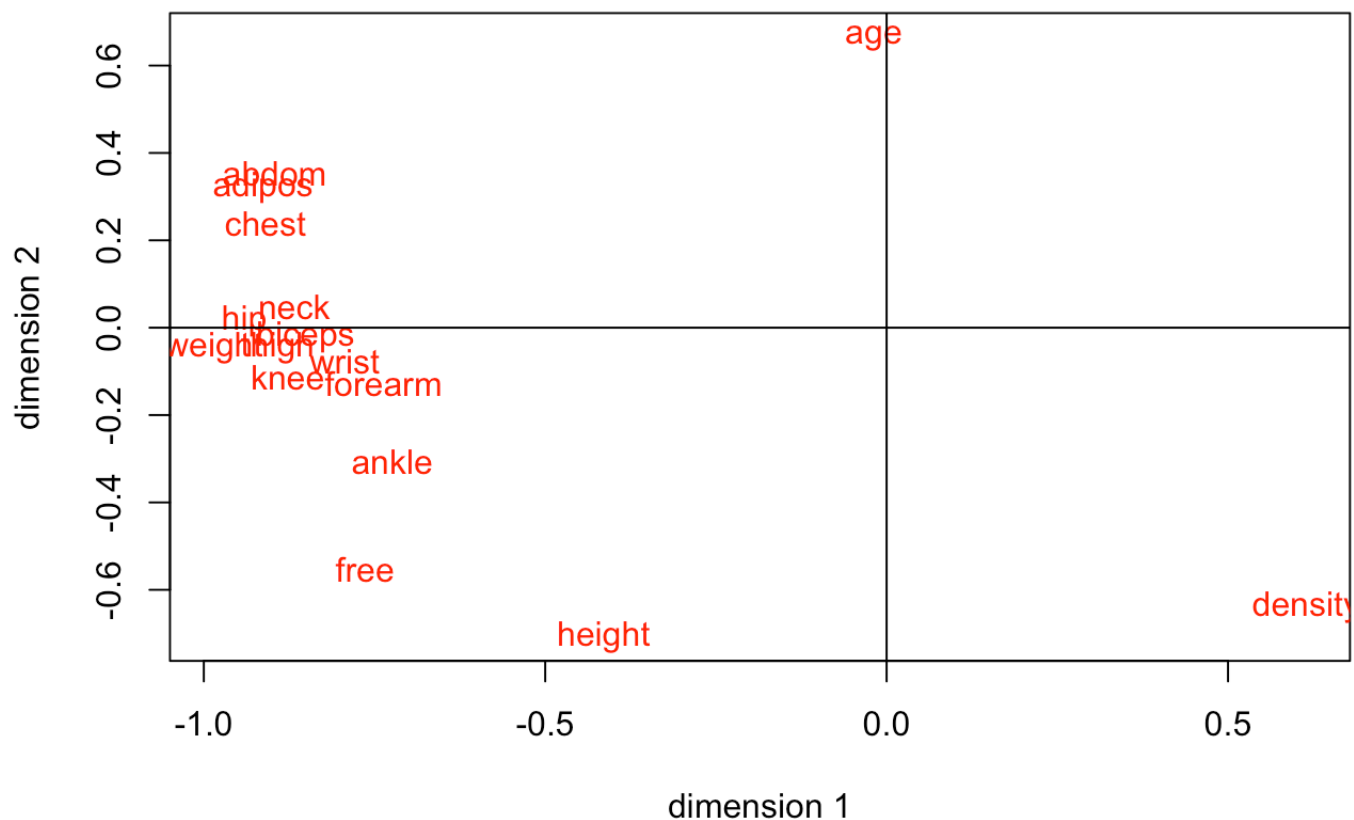
```
## Iteration:    1 fold:    11.93445666 fnew:    12.23316284
## Iteration:    2 fold:    12.23316284 fnew:    12.24922704
## Iteration:    3 fold:    12.24922704 fnew:    12.25509701
## Iteration:    4 fold:    12.25509701 fnew:    12.25870256
## Iteration:    5 fold:    12.25870256 fnew:    12.26121744
## Iteration:    6 fold:    12.26121744 fnew:    12.26300909
## Iteration:    7 fold:    12.26300909 fnew:    12.26429152
## Iteration:    8 fold:    12.26429152 fnew:    12.26521099
## Iteration:    9 fold:    12.26521099 fnew:    12.26587082
## Iteration:   10 fold:    12.26587082 fnew:    12.26634466
## Iteration:   11 fold:    12.26634466 fnew:    12.26668514
## Iteration:   12 fold:    12.26668514 fnew:    12.26692993
## Iteration:   13 fold:    12.26692993 fnew:    12.26710602
## Iteration:   14 fold:    12.26710602 fnew:    12.26723274
## Iteration:   15 fold:    12.26723274 fnew:    12.26732397
## Iteration:   16 fold:    12.26732397 fnew:    12.26738968
## Iteration:   17 fold:    12.26738968 fnew:    12.26743702
## Iteration:   18 fold:    12.26743702 fnew:    12.26747113
## Iteration:   19 fold:    12.26747113 fnew:    12.26749573
## Iteration:   20 fold:    12.26749573 fnew:    12.26751346
## Iteration:   21 fold:    12.26751346 fnew:    12.26752624
## Iteration:   22 fold:    12.26752624 fnew:    12.26753546
## Iteration:   23 fold:    12.26753546 fnew:    12.26754212
## Iteration:   24 fold:    12.26754212 fnew:    12.26754692
## Iteration:   25 fold:    12.26754692 fnew:    12.26755038
## Iteration:   26 fold:    12.26755038 fnew:    12.26755288
## Iteration:   27 fold:    12.26755288 fnew:    12.26755468
## Iteration:   28 fold:    12.26755468 fnew:    12.26755598
## Iteration:   29 fold:    12.26755598 fnew:    12.26755692
```

The eigenvalues of the correlation matrix, in percentages, with a plot of their cumulative values, are as follows.

```
## [1] 0.6369047237 0.1298175838 0.0593329100 0.0399460439 0.0308797444
## [6] 0.0285711710 0.0181510533 0.0148222809 0.0128746126 0.0110294094
## [11] 0.0076337401 0.0042483034 0.0026208110 0.0021778757 0.0006752046
## [16] 0.0003145321
```



The scatterplot of the first two principal components is as follows.



Example 3: Air Pollution

The Sokal/Rolph air pollution data have previously been used by Wang and Murphy (2004) to illustrate ACE (Breiman and Friedman 1985). The outcome variable is sulphur dioxide concentration in air mgs. per cubic metre in 41 cities in the USA. The predictors are

- average annual temperature in degrees F
- number of manufacturers employing >20 workers
- population size in thousands
- average annual wind speed in miles per hour
- average annual rainfall in inches
- average number of days rainfall per year

```
data(usair, package = "gamlss.data")
usair_knots <- lapply (usair, function (x) fivenum (x)[2:4])
usair_degrees <- c (3,3,3,3,3,3,3)
usair_ordinal <- c(TRUE,TRUE,TRUE,TRUE,FALSE,TRUE,FALSE)
```

```
airsmc<-aspect(usair,usair_knots,usair_degrees,usair_ordinal,verbose=0, itmax = 1000,
maxsmc,p=1)
```

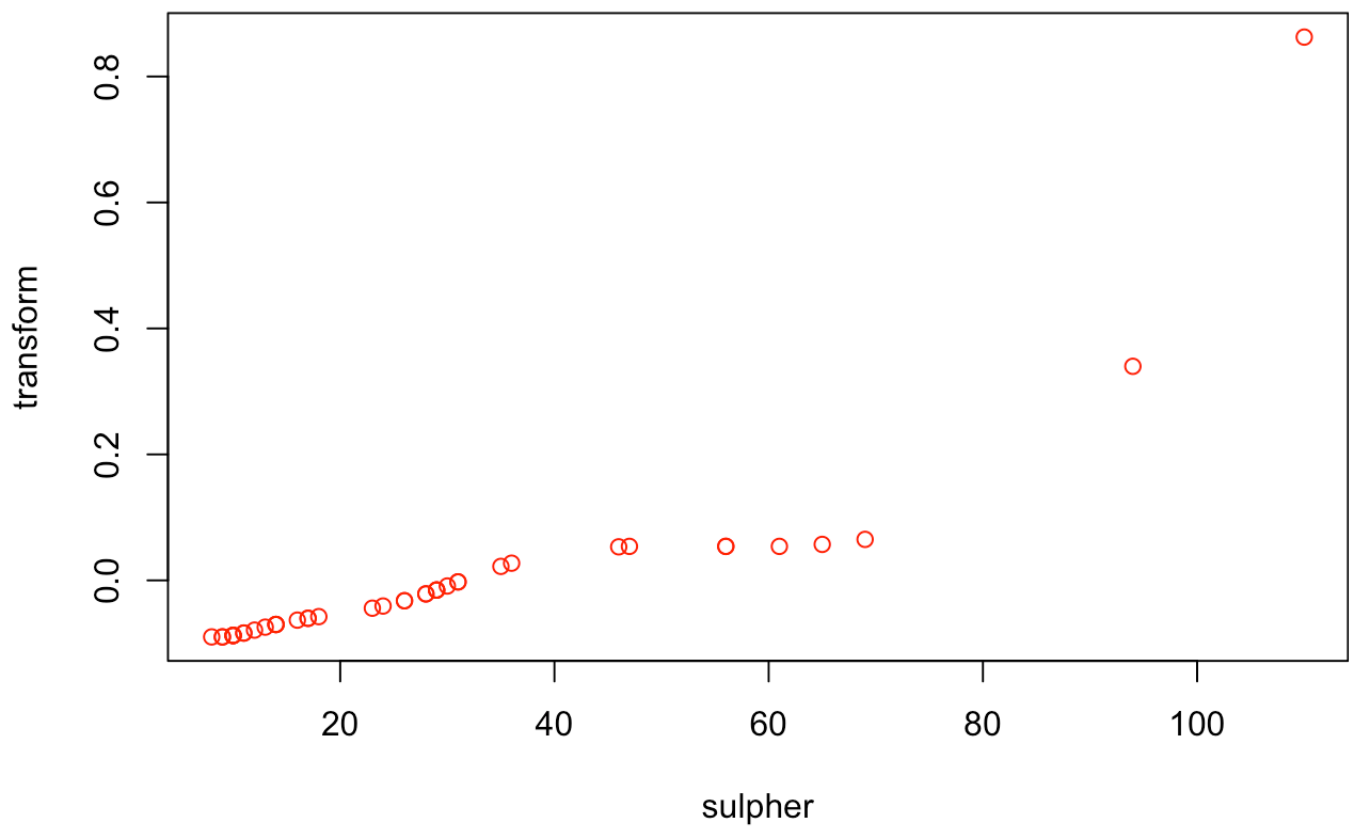
The optimal SMC is 0.9482315 (for ACE this is 0.9469), the correlation matrix is

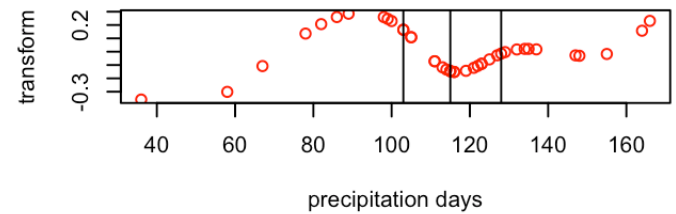
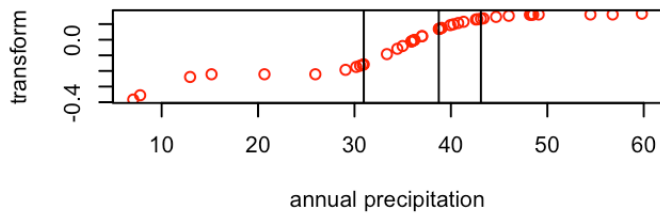
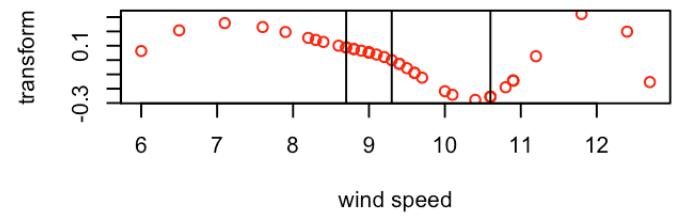
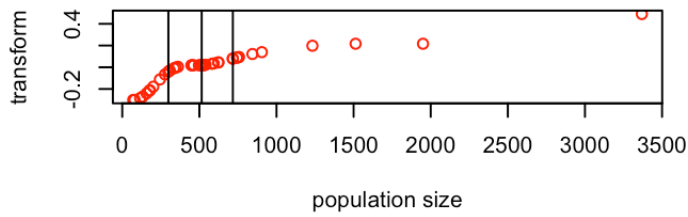
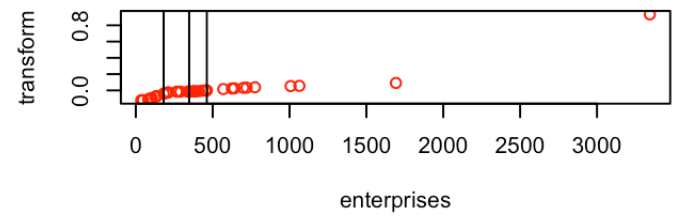
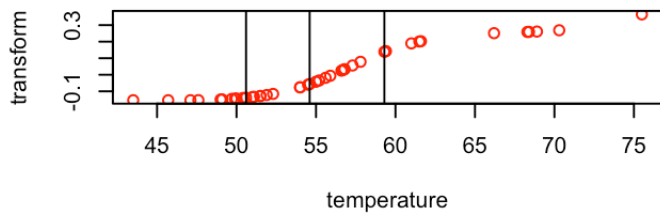
##	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
## [1,]	1.0000000	-0.34401533	0.85100005	0.34248610	-0.4040813	0.04775350
## [2,]	-0.3440153	1.00000000	-0.16310443	0.06765795	0.1208380	0.34046001
## [3,]	0.8510000	-0.16310443	1.00000000	0.69790740	-0.3560280	-0.03922927
## [4,]	0.3424861	0.06765795	0.69790740	1.00000000	-0.3947447	-0.05441145
## [5,]	-0.4040813	0.12083797	-0.35602805	-0.39474473	1.0000000	0.07234770
## [6,]	0.0477535	0.34046001	-0.03922927	-0.05441145	0.0723477	1.00000000
## [7,]	-0.1707223	-0.24678998	-0.09406506	-0.13570731	-0.1345694	-0.05539428
##	[,7]					
## [1,]	-0.17072228					
## [2,]	-0.24678998					
## [3,]	-0.09406506					
## [4,]	-0.13570731					
## [5,]	-0.13456940					
## [6,]	-0.05539428					
## [7,]	1.00000000					

and the regression coefficients are

## [1]	-0.2058751	1.0722221	-0.5066085	-0.2361394	0.1375961	-0.2135771
--------	------------	-----------	------------	------------	-----------	------------

The seven transformation plots are as follows.





We also use this example to show that an empty vector of interior knots leads to a polynomial transformation. In this case we transform the dependent variable linearly (which basically means we do not transform it at all).

```
usair_knots[[1]]<-numeric(0)
usair_degrees[1]<-1
airlin<-aspect(usair,usair_knots,usair_degrees,usair_ordinal,verbose=0,maxsmc,p=1)
```

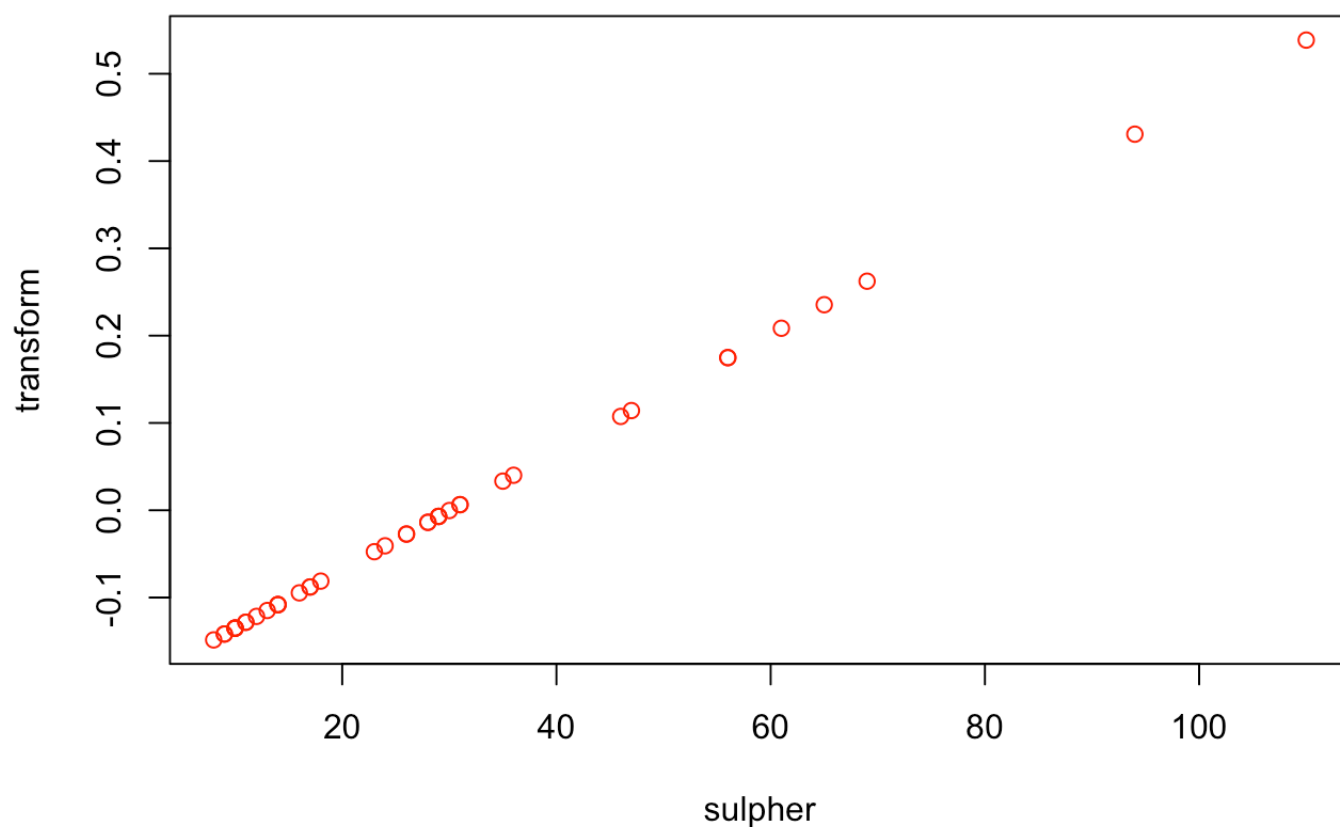
The optimal SMC is now 0.904187, the correlation matrix is

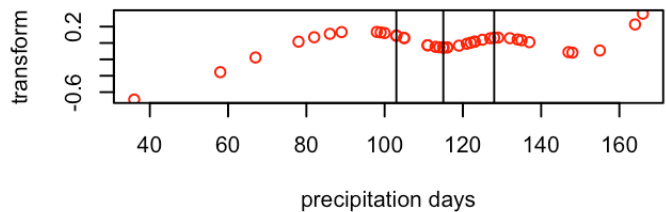
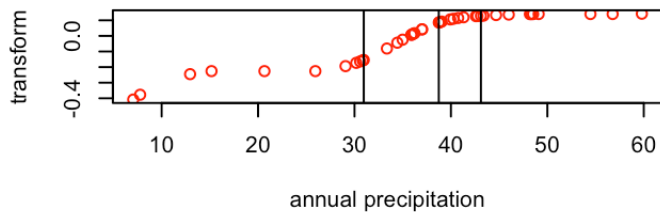
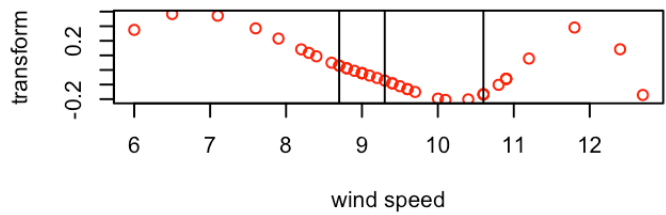
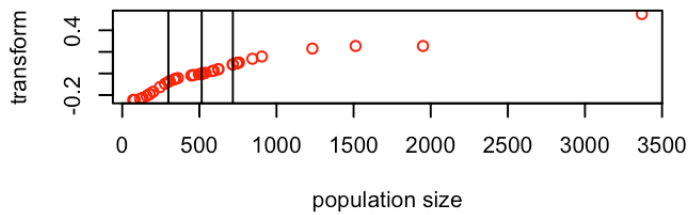
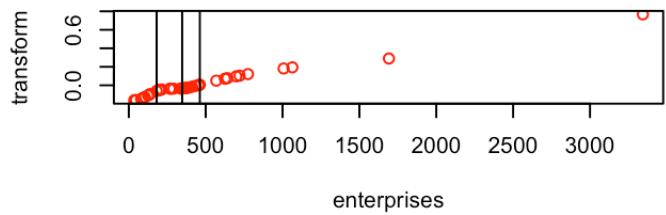
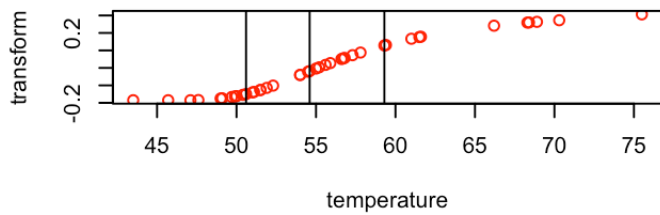
```
##          [,1]      [,2]      [,3]      [,4]      [,5]
## [1,]  1.000000000 -0.4829229  0.60609579  0.28419105 -0.36590731
## [2,] -0.482922926  1.0000000 -0.19383688  0.01929680  0.15465911
## [3,]  0.606095786 -0.1938369  1.00000000  0.86739171 -0.35180667
## [4,]  0.284191054  0.0192968  0.86739171  1.00000000 -0.37329140
## [5,] -0.365907315  0.1546591 -0.35180667 -0.37329140  1.00000000
## [6,]  0.175496105  0.2817608 -0.02235636 -0.03976988 -0.01936234
## [7,]  0.002801162 -0.3629289  0.05071848 -0.03275958 -0.22797737
##          [,6]      [,7]
## [1,]  0.17549611  0.002801162
## [2,]  0.28176080 -0.362928895
## [3,] -0.02235636  0.050718480
## [4,] -0.03976988 -0.032759583
## [5,] -0.01936234 -0.227977370
## [6,]  1.00000000  0.359290635
## [7,]  0.35929063  1.000000000
```

and the regression coefficients are

```
## [1] -0.5412443  1.1010932 -0.7656825 -0.2898311  0.5039996 -0.5217185
```

The seven transformation plots are as follows.





Example 4: Moral Integration of American Cities

The `angell` data frame has 43 rows and 4 columns. The observations are 43 U. S. cities around 1950. These data were previously analyzed with ACE by De Veaux (1987). The variables are

- Moral Integration: Composite of crime rate and welfare expenditures.
- Ethnic Heterogeneity: From percentages of nonwhite and foreign-born white residents.
- Geographic Mobility: From percentages of residents moving into and out of the city.
- A factor with levels: Northeast; Midwest; Southeast; West.

```
data(Angell, package = "car")
angell <- Angell[,1:3]
angell_knots <- lapply (angell, function (x) fivenum (x)[2:4])
angell_degrees <- c(2,2,2)
angell_ordinal <- c(TRUE, TRUE, TRUE)
```

```
hsmc<-aspect(angell,angell_knots,angell_degrees,angell_ordinal,maxsmc,p=1)
```

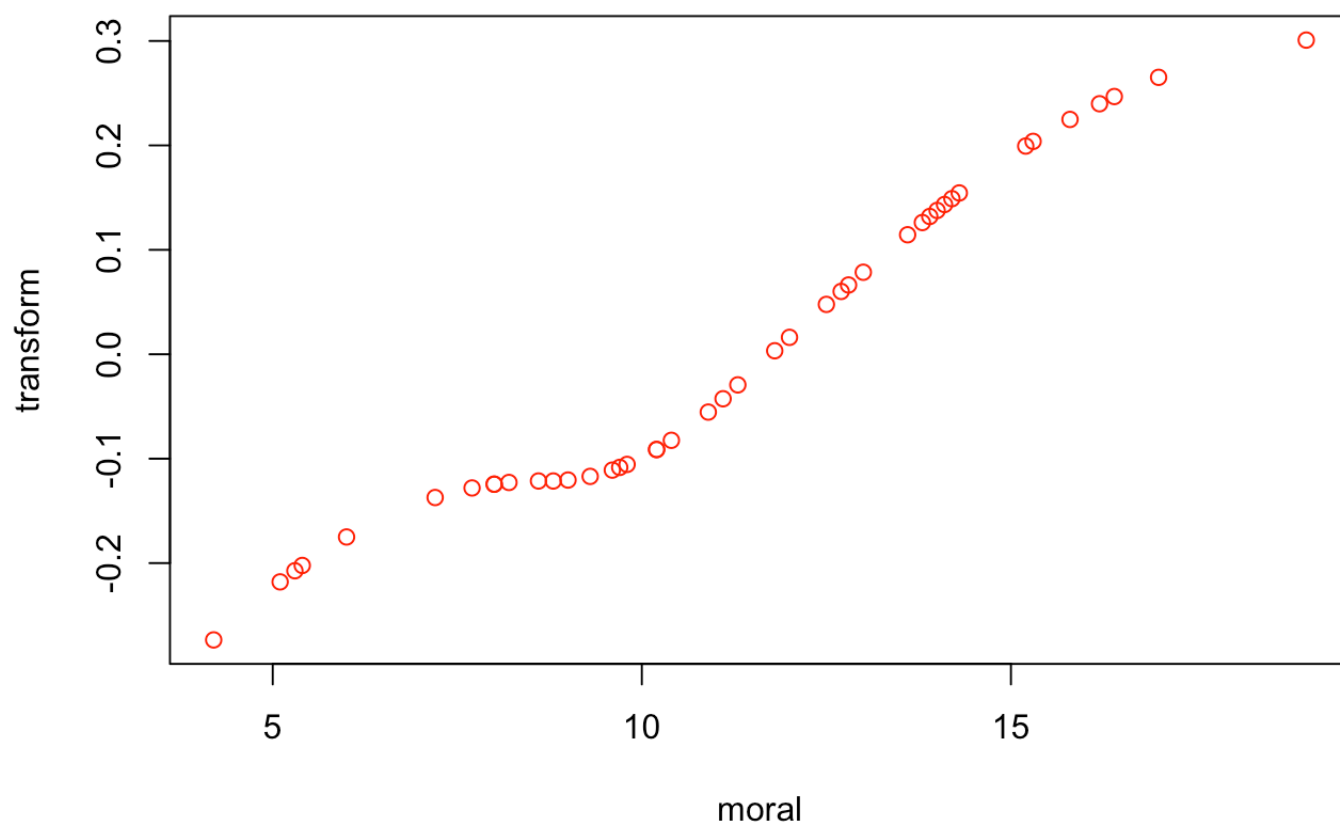


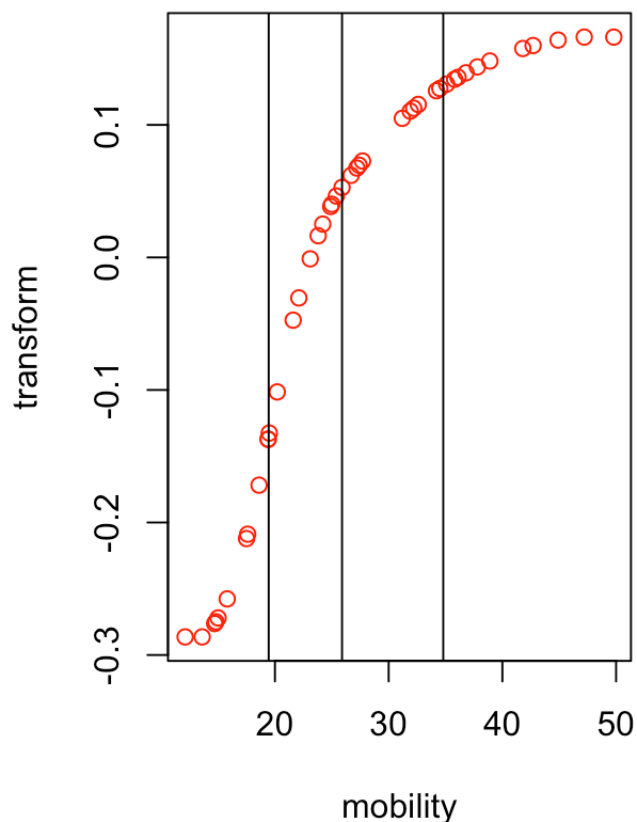
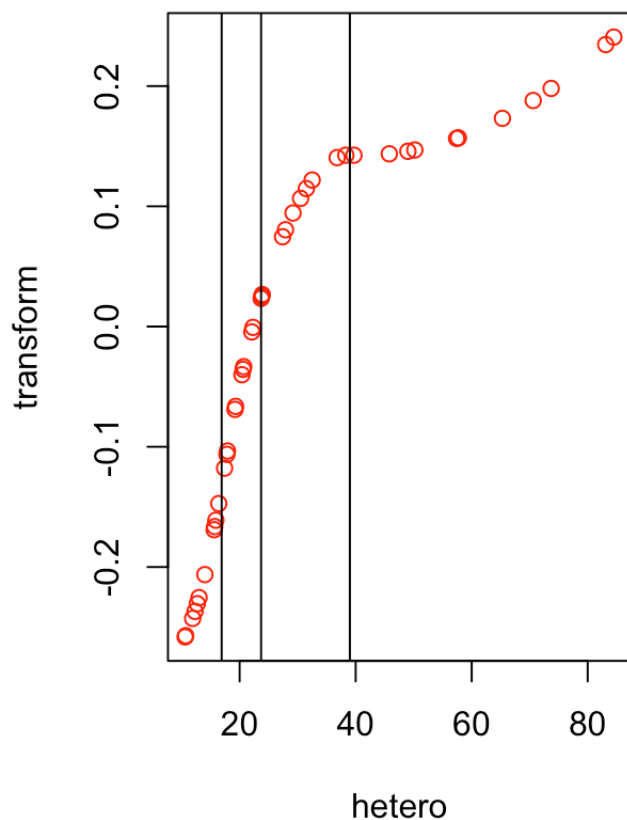
```
## Iteration: 1 fold: 0.67555675 fnew: 0.73923854
## Iteration: 2 fold: 0.73923854 fnew: 0.74680608
## Iteration: 3 fold: 0.74680608 fnew: 0.74931438
## Iteration: 4 fold: 0.74931438 fnew: 0.75005255
## Iteration: 5 fold: 0.75005255 fnew: 0.75026068
## Iteration: 6 fold: 0.75026068 fnew: 0.75031127
## Iteration: 7 fold: 0.75031127 fnew: 0.75032348
## Iteration: 8 fold: 0.75032348 fnew: 0.75032642
## Iteration: 9 fold: 0.75032642 fnew: 0.75032713
```

Correlations between transformed variables are as follows.

```
##          [,1]      [,2]      [,3]
## [1,] 1.0000000 -0.53934872 -0.64048623
## [2,] -0.5393487 1.00000000 -0.06643057
## [3,] -0.6404862 -0.06643057 1.00000000
```

The transformation plots are as follows.





Example 5: Thirteen Personality Scales

We illustrate a more complicated aspect using a small data set from the `psych` package (Revelle 2015) of five scales from the Eysenck Personality Inventory, five from a Big 5 inventory, a Beck Depression Inventory, and State and Trait Anxiety measures.

```
data(epi.bfi, package = "psych")
epi <- epi.bfi
epi_knots <- lapply (epi, function (x) fivenum (x)[2:4])
epi_degrees <- rep (2, 13)
epi_ordinal <- rep (TRUE, 13)
```

The aspect we use is the maximum log-likelihood of a factor analysis model. Because this aspect is the maximum of a family of functions linear in $\mathbf{R}$ it is indeed convex. Obviously the same result applies to any structural equation model combined with a multinormal log-likelihood. Using the negative logarithm of the determinant as an aspect corresponds with the special case of a saturated model.

```

maxfac <- function (r, p) {
  fa <- factanal (NULL, p, covmat = r, rotation = "none")
  s <- tcrossprod (fa$loadings) + diag (fa$unique)
  g <- - solve (s)
  f <- sum (g * r) - log(det (s))
  return (list (f = f, g = g))
}

```

We fit a two-factor model, without worrying if two factors are appropriate for these data.

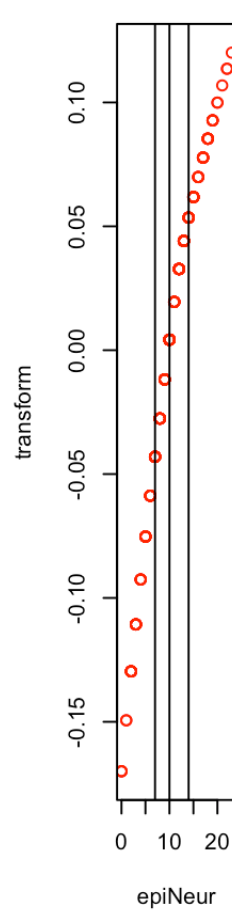
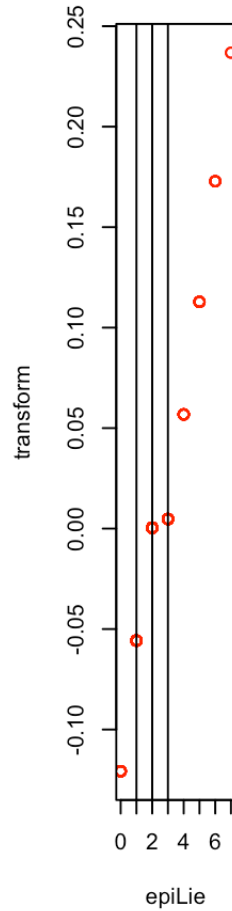
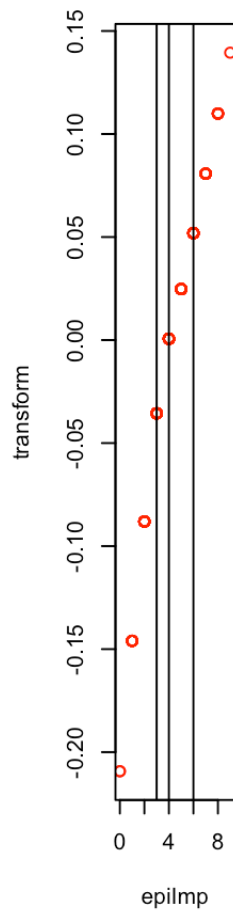
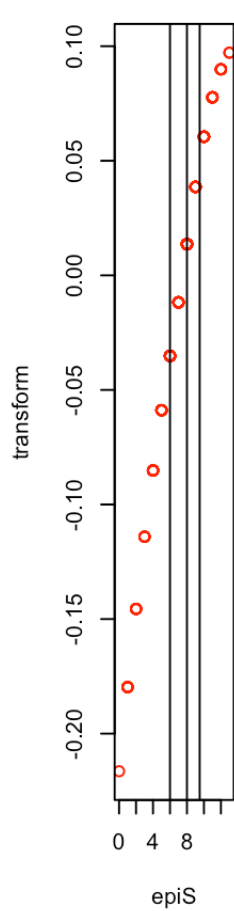
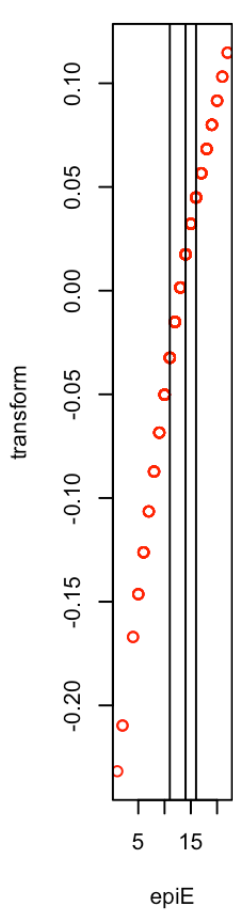
```
efac<-aspect(emi,emi_knots,emi_degrees,emi_ordinal,maxfac,p=2)
```

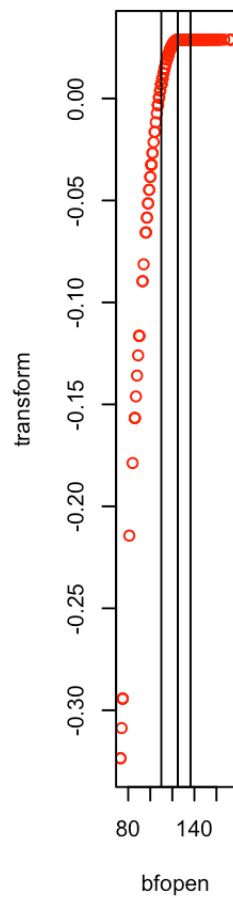
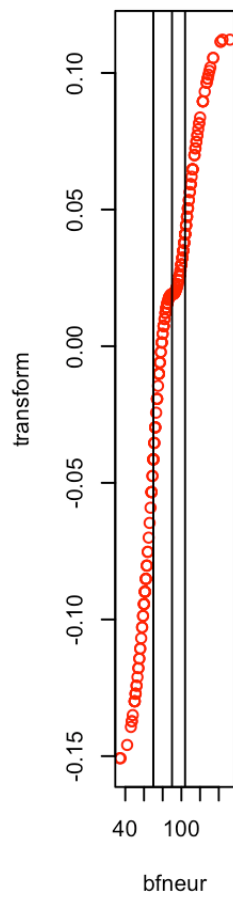
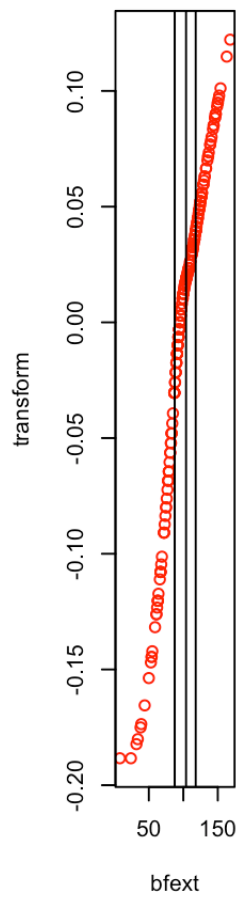
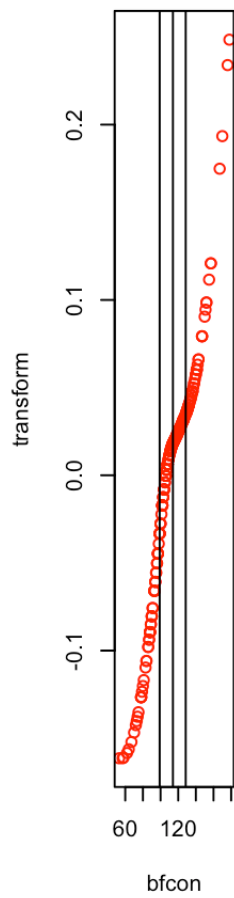
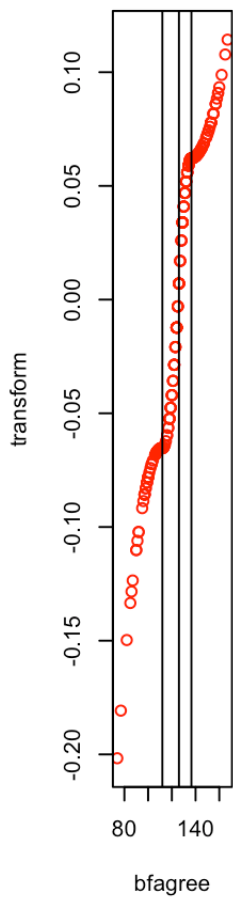
```

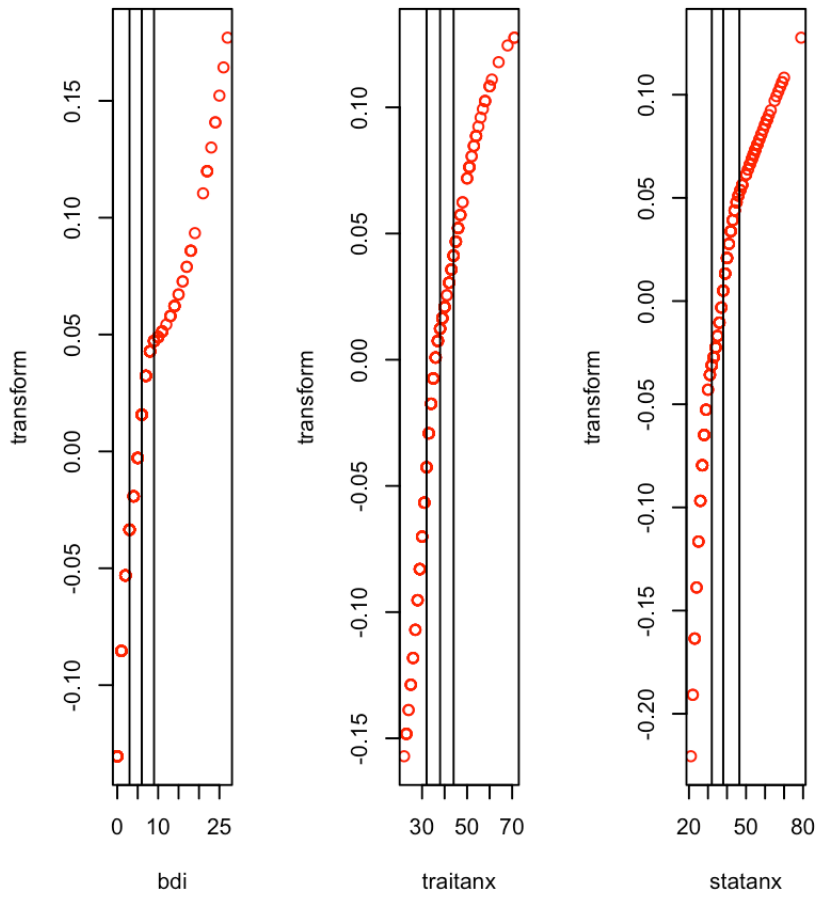
## Iteration:    1 fold:    -7.47534413 fnew:    -7.08355786
## Iteration:    2 fold:    -7.08355786 fnew:    -7.05118236
## Iteration:    3 fold:    -7.05118236 fnew:    -7.03813254
## Iteration:    4 fold:    -7.03813254 fnew:    -7.03277037
## Iteration:    5 fold:    -7.03277037 fnew:    -7.03050114
## Iteration:    6 fold:    -7.03050114 fnew:    -7.02952915
## Iteration:    7 fold:    -7.02952915 fnew:    -7.02911088
## Iteration:    8 fold:    -7.02911088 fnew:    -7.02893056
## Iteration:    9 fold:    -7.02893056 fnew:    -7.02885277
## Iteration:   10 fold:    -7.02885277 fnew:    -7.02881920
## Iteration:   11 fold:    -7.02881920 fnew:    -7.02880472
## Iteration:   12 fold:    -7.02880472 fnew:    -7.02879847
## Iteration:   13 fold:    -7.02879847 fnew:    -7.02879577
## Iteration:   14 fold:    -7.02879577 fnew:    -7.02879461
## Iteration:   15 fold:    -7.02879461 fnew:    -7.02879411

```

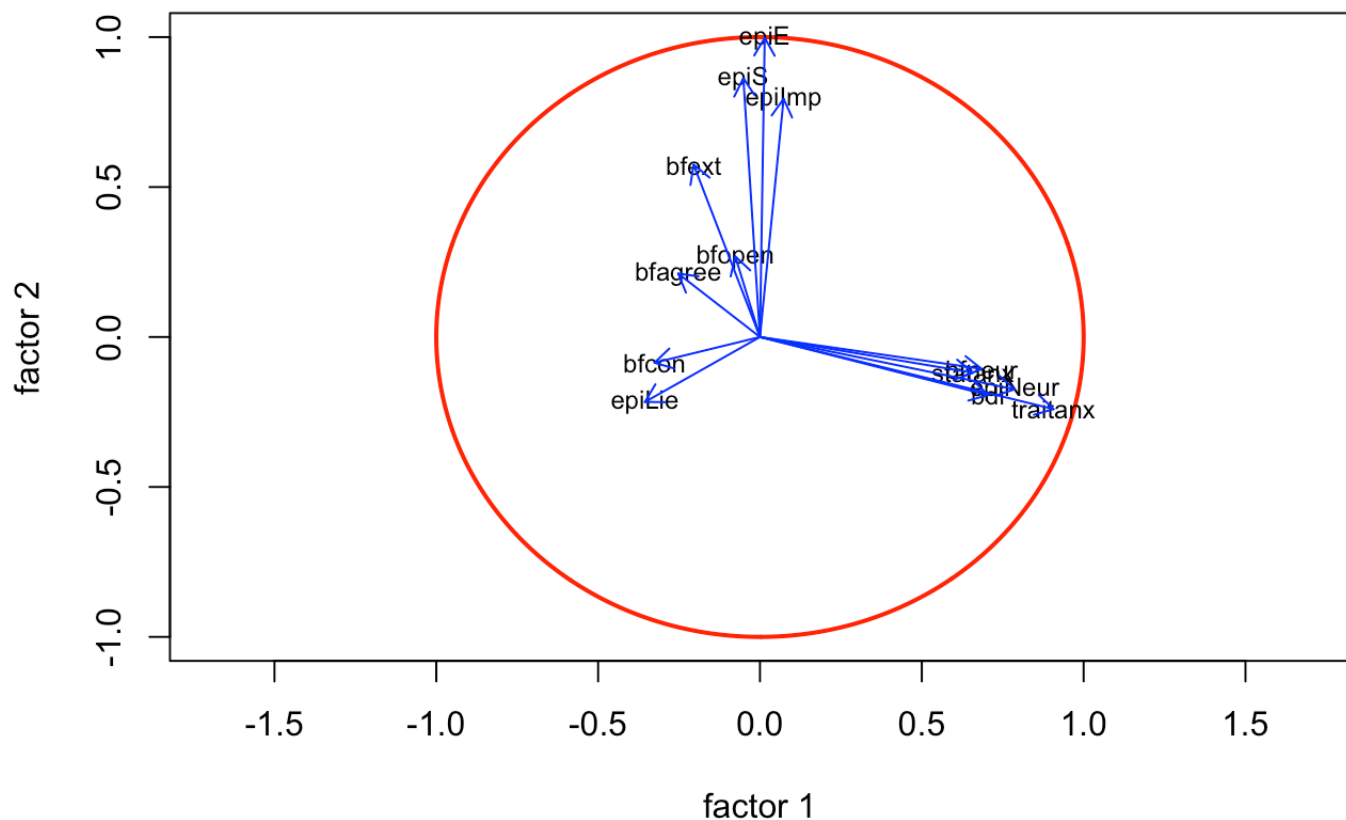
The transformation plots are as follows.







Here are the factor loadings, for those so inclined.



And here are the uniquenesses. Variable 1 (epiE) seems to be heading for Heywood.

##	[1]	0.0050000	0.2527709	0.3638904	0.8268849	0.3507721	0.8915067	0.8864692
##	[8]	0.6280157	0.5196517	0.9219710	0.4660321	0.1194840	0.5526974	

References

Breiman, L., and J. H. Friedman. 1985. "Estimating Optimal Transformations for Multiple Regression and Correlation." *Journal of the American Statistical Association* 80: 580–619.

De Leeuw, J. 1988a. "Multivariate Analysis with Linearizable Regressions." *Psychometrika* 53: 437–54.
http://www.stat.ucla.edu/~deleeuw/janspubs/1988/articles/deleeuw_A_88a.pdf
 (http://www.stat.ucla.edu/~deleeuw/janspubs/1988/articles/deleeuw_A_88a.pdf).

— — —. 1988b. "Multivariate Analysis with Optimal Scaling." In *Proceedings of the International Conference on Advances in Multivariate Statistical Analysis*, edited by S. Das Gupta and J.K. Ghosh, 127–60. Calcutta, India: Indian Statistical Institute. http://www.stat.ucla.edu/~deleeuw/janspubs/1988/chapters/deleeuw_C_88b.pdf
 (http://www.stat.ucla.edu/~deleeuw/janspubs/1988/chapters/deleeuw_C_88b.pdf).

— — —. 2014. "History of Nonlinear Principal Component Analysis." In *The Visualization and Verbalization of Data*, edited by J. Blasius and M. Greenacre. Chapman; Hall.
http://www.stat.ucla.edu/~deleeuw/janspubs/2014/chapters/deleeuw_C_14.pdf
 (http://www.stat.ucla.edu/~deleeuw/janspubs/2014/chapters/deleeuw_C_14.pdf).

- — —. 2015a. "Block Relaxation Algorithms in Statistics, Part I." <http://jandeleeuw.gitbooks.io/bras1/content/> (<http://jandeleeuw.gitbooks.io/bras1/content/>).
- — —. 2015b. "Block Relaxation Algorithms in Statistics, Part II." <http://jandeleeuw.gitbooks.io/bras2/content/> (<http://jandeleeuw.gitbooks.io/bras2/content/>).
- — —. 2015c. "Regression with Linear Inequality Restrictions on Predicted Values." <http://rpubs.com/deleeuw/78897> (<http://rpubs.com/deleeuw/78897>).
- — —. 2015d. "Exceedingly Simple B-spline Code." <http://rpubs.com/deleeuw/79161> (<http://rpubs.com/deleeuw/79161>).
- — —. 2015e. "Exceedingly Simple Gram-Schmidt Code." <http://rpubs.com/deleeuw/84866> (<http://rpubs.com/deleeuw/84866>).
- De Veaux, R.D. 1987. "Finding Transformations for Regression Using the ACE Algorithm." *Sociological Methods and Research* 18: 327–59.
- Gifi, A. 1990. *Nonlinear Multivariate Analysis*. New York, N.Y.: Wiley.
- Koyak, R. 1987. "On Measuring Internal Dependence in a Set of Random Variables." *Annals of Statistics* 15: 1215–28.
- Kruskal, J. B. 1965. "Analysis of Factorial Experiments by Estimating Monotone Transformations of the Data." *Journal of the Royal Statistical Society B27*: 251–63.
- Linting, M., J.J. Meulman, P.J.F. Groenen, and A.J. Van Der Kooij. 2007. "Nonlinear Principal Components Analysis: Introduction and Application." *Psychological Methods* 12: 336–58.
- Mair, P., and J. De Leeuw. 2010. "A General Framework for Multivariate Analysis with Optimal Scaling: The R Package aspect." *Journal of Statistical Software* 32 (9): 1–23.
http://www.stat.ucla.edu/~deleeuw/janspubs/2010/articles/mair_deleeuw_A_10.pdf
(http://www.stat.ucla.edu/~deleeuw/janspubs/2010/articles/mair_deleeuw_A_10.pdf).
- Mullen, K.M., and I. H. M. van Stokkum. 2012. *Nnls: The Lawson-Hanson Algorithm for Non-Negative Least Squares (NNLS)*. <http://CRAN.R-project.org/package=nnls> (<http://CRAN.R-project.org/package=nnls>).
- Revelle, William. 2015. *Psych: Procedures for Psychological, Psychometric, and Personality Research*. Evanston, Illinois: Northwestern University. <http://CRAN.R-project.org/package=psych> (<http://CRAN.R-project.org/package=psych>).
- Van Der Kooij, A.J., J.J. Meulman, and W.J. Heiser. 2007. "Local Minima in Categorical Multiple Regression." *Computational Statistics and Data Analysis* 50: 446–62.
- Wang, D., and M. Murphy. 2004. "Estimating Optimal Transformations for Multiple Regression Using the ACE Algorithm." *Journal of Data Science* 2: 329–46.
- Winsberg, S., and J. O. Ramsay. 1980. "Monotone Transformations to Additivity." *Biometrika* 67: 669–74.
- — —. 1983. "Monotone Spline Transformations for Dimension Reduction." *Psychometrika* 48: 575–95.
- Young, F.W., J. De Leeuw, and Y. Takane. 1976. "Regression with Qualitative and Quantitative Data: An Alternating Least Squares Approach with Optimal Scaling Features." *Psychometrika* 41: 505–29.
http://www.stat.ucla.edu/~deleeuw/janspubs/1976/articles/young_deleeuw_takane_A_76.pdf

(http://www.stat.ucla.edu/~deleeuw/janspubs/1976/articles/young_deleeuw_takane_A_76.pdf).

Young, F.W., Y. Takane, and J. De Leeuw. 1978. "The Principal Components of Mixed Measurement Level Multivariate Data: an Alternating Least Squares Method with Optimal Scaling Features." *Psychometrika* 45: 279–81. http://www.stat.ucla.edu/~deleeuw/janspubs/1978/articles/young_takane_deleeuw_A_78.pdf (http://www.stat.ucla.edu/~deleeuw/janspubs/1978/articles/young_takane_deleeuw_A_78.pdf).